О некоторых алгоритмах работы с длинными строками и их применении в задачах дискретной оптимизации

Александр Панин
ag.panin@gmail.com

Тольятти-2011

Актуальность темы

Актуальность темы обуславливается следующими факторами:

- Широкая область применения дискретной оптимизации и методов обработки строк;
- Высокая алгоритмическая сложность задач из указанных областей;
- Отсутствие точных алгоритмов, работающих за полиномиальное время, для большинства задач.

Цель работы

Целью работы является разработка быстрых алгоритмов сравнения длинных строк, а так же разработка подхода к оценке качества разработанных алгоритмов.

Основные задачи

- Разработка оригинальной метрики на множестве дискретных сигналов на основе мультиэвристического подхода к задачам дискретной оптимизации;
- Разработка и реализация алгоритма кластеризации сигналов акустической эмиссии на основе разработанной метрики;
- Разработка специальной метрики на множестве символьных строк на основе мультиэвристического подхода к задачам дискретной оптимизации. Получение – в зависимости от конкретных применяемых эвристик – нового подхода к определению расстояния между строками, а также быстрых алгоритмов для аппроксимации расстояния Лёвенштейна или выравнивания Нидлмана-Вунша;
- Применение разработанных метрик для создания и реализации алгоритмов сравнения генетических последовательностей;
- Разработка и реализация подхода к оценке эффективности (качества) эвристических алгоритмов.

Научная новизна

- Разработаны алгоритмы сравнения дискретных сигналов и генетических последовательностей на основе мультиэвристического подхода к задачам дискретной оптимизации. Данные алгоритмы задают «адекватные» метрики, позволяющие выделять объекты по их существенным признакам. К тому же алгоритмы являются достаточно быстрыми, в среднем время их работы близко к линейному.

Научная новизна

- Предложен подход к сравнению эффективности (качества) эвристических алгоритмов, основанный на кластеризации. Данный подход позволяет оценить способность метрики разделять объекты на группы по их существенным признакам.

Научная новизна

- Разработаны алгоритмы точного и приближённого поиска наибольшей общей подпоследовательности на основе метода динамического программирования. Точный алгоритм работает быстрее всех известных автору алгоритмов, требования к памяти одни из самых низких, на уровне других быстрых алгоритмов. Алгоритм приближённого поиска работает на порядки быстрее алгоритма точного поиска – за линейное время. Требования к памяти такие же, точность – порядка 0,99 для фрагментов ДНК длиной до 1 млн символов.

Применение мульэвристического подхода для анализа сигналов акустической эмиссии

Математическая модель сигнала

- $$s(t) = \left(x_{t_1}, x_{t_2}, \dots, x_{t_n}\right),$$
$$t_i, n \in \mathbb{N}, x_{t_i} \in \mathbb{Z}, \Delta = t_{i+1} - t_i = \text{const}$$

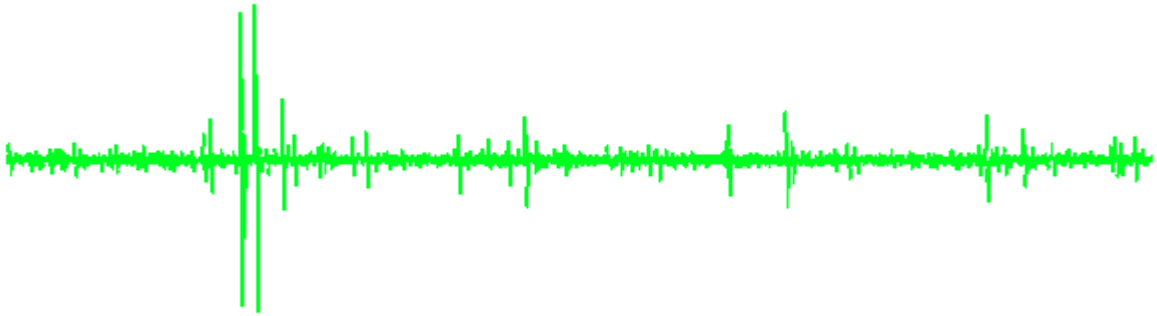
Этапы анализа

Чтение сигналов

Вейвлет-преобразование

Сравнение

Кластеризация



Применение мультиэвристического подхода

Эвристики:

- Для небольших фрагментов сигналов, начинающихся с текущей позиции, выполняется специальная версия алгоритма вычисления расстояния Лёвенштейна (этот алгоритм будет описан ниже). По построенной этим алгоритмом таблице определяется оптимальная траектория сдвига.
- Ближайшие позиции оцениваются исходя из удалённости исходной позиции и разности между значениями сигналов, сдвиг происходит в позицию с наибольшей оценкой.

Применение мультиэвристического подхода

Эвристики:

- Для небольших фрагментов сигналов, начинающихся с текущей позиции, выполняется специальная версия алгоритма вычисления расстояния Лёвенштейна (этот алгоритм будет описан ниже). По построенной этим алгоритмом таблице определяется оптимальная траектория сдвига.
- Ближайшие позиции оцениваются исходя из удалённости исходной позиции и разности между значениями сигналов, сдвиг происходит в позицию с наибольшей оценкой.

Другие способы задания параметров кластеризации

- Функция корреляции
- Специальная версия алгоритма вычисления расстояния Лёвенштейна для сравнения сигналов
- Параметры акустической эмиссии

Функция корреляции

- $$f(x,y) = \frac{1}{N}\sum_{i=1}^{N} \cos\alpha(x_i, y_i)$$

$$\cos\alpha(x_i, y_i) = \frac{a_{x_i}a_{y_i} + b_{x_i}b_{y_i}}{\sqrt{(a_{x_i}{}^2 + b_{x_i}{}^2)(a_{y_i}{}^2 + b_{y_i}{}^2)}},$$

$$b_{x_i} = x_i - x_{i-1},\ b_{y_i} = y_i - y_{i-1},\ a_{x_i} = a_{y_i} = dt,$$

Специальная версия алгоритма вычисления расстояния Лёвенштейна

$c(x, y) = D(n, m) / max(n, m)$, где

$$D(i, j) = \begin{cases} 0, & \text{если } i = 0 \text{ или } j = 0 \\ max(\ D(i-1, j-1) + correlation(x_i, y_j), & \text{в остальных случаях} \\ D(i-1, j) + penalty, D(i, j-1)) - penalty) & \end{cases}$$

Специальная версия алгоритма вычисления расстояния Лёвенштейна

Пусть имеются следующие сигналы:

- x = [532,594; 109,611; -1998,57; 1739,08; -1918,29; -2183,42; 5037,88; -4283,07; 1282,17; 2613,6],
- y = [-377,147; 1634,49; 585,875; -4424,19; 5364,67; -4999,32; 2634,9; -279,988; 3836,9; -2502,25].

Специальная версия алгоритма вычисления расстояния Лёвенштейна

	1	2	3	4	5	6	7	8	9	10
1	0,40	0,40	0,95	0,54	-0,10	0,40	-0,05	0,69	0,11	0,48
2	0,40	0,40	0,95	0,54	-0,10	0,40	-0,05	0,69	0,11	0,48
3	-0,25	-0,25	0,94	0,94	-0,69	0,88	-0,66	0,99	-0,52	0,92
4	0,96	0,96	-0,17	-0,76	0,97	-0,85	0,98	-0,62	1,00	-0,80
5	-0,47	-0,47	0,84	0,99	-0,84	0,97	-0,81	1,00	-0,71	0,99
6	0,49	0,49	0,91	0,46	0,00	0,31	0,04	0,62	0,20	0,40
7	0,90	0,90	-0,35	-0,86	1,00	-0,93	1,00	-0,75	0,99	-0,89
8	-0,67	-0,67	0,68	0,99	-0,95	1,00	-0,93	0,95	-0,86	1,00
9	0,92	0,92	-0,30	-0,83	0,99	-0,91	1,00	-0,71	1,00	-0,86
10	0,98	0,98	0,29	-0,38	0,76	-0,51	0,78	-0,19	0,87	-0,43

Параметры акустической эмиссии

- число, частота и энергия импульсов сигнала АЭ, превышающих пороговый уровень, за единицу времени;
- число, частота и энергия всех импульсов сигнала АЭ за единицу времени.

Кластеризация

- Для заданного множества K входных векторов x_k и N выделяемых кластеров c_j предполагается, что любой x_k принадлежит любому c_j с принадлежностью u_{jk} интервалу [0,1], где j – номер кластера, а k – номер входного вектора.

Принимаются во внимание следующие условия нормирования для u_{jk}:

$$\sum_{j=1}^{N} u_{jk} = 1, \forall k = 1, \ldots, K;$$

$$0 < \sum_{j=1}^{N} u_{jk} \leq K, \forall j = 1, \ldots, N.$$

Кластеризация

- $$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}{}^{q} \left\| x_k - c_j \right\| \to min,$$

где q – фиксированный параметр алгоритма.

Кластеризация

- $$\frac{d}{du_{jk}}\left(\sum_{j=1}^{N}\sum_{k=1}^{K} {u_{jk}}^{q} \left\|x_k - c_j\right\|\right) = 0,$$

$$\frac{d}{dc_j}\left(\sum_{j=1}^{N}\sum_{k=1}^{K} {u_{jk}}^{q} \left\|x_k - c_j\right\|\right) = 0.$$

Кластеризация

- $$c_j = \frac{\sum_{j=1}^{N} u_{jk}{}^{q} \cdot x_k}{\sum_{k=1}^{K} u_{jk}{}^{q}}$$

$$u_{jk} = \frac{\frac{1}{\left\| x_k - c_j \right\|^{1/q-1}}}{\sum_{j=1}^{N} \frac{1}{\left\| x_k - c_j \right\|^{1/q-1}}}.$$

Результаты

1		2		3		4		5		6	
кластер 1	кластер 2	кластер 1	кластер 2	кластер 1	кластер 2	кластер 1	кластер 2	кластер 1	кластер 2	кластер 1	кластер 2
0,816172	0,183828	0,0841623	0,915838	0,0505967	0,949403	0,790016	0,209984	0,224708	0,775292	0,885976	0,114024
0,113145	0,886855	0,103572	0,896428	0,868207	0,131793	0,017823	0,982177	0,492051	0,507949	0,0614656	0,938534
0,0866124	0,913388	0,054635	0,945365	0,131692	0,868308	0,0338249	0,966175	0,243453	0,756547	0,0165942	0,983406
0,0357886	0,964211	0,1069	0,8931	0,79034	0,20966	0,0119785	0,988021	0,205172	0,794828	0,0642309	0,935769
0,681351	0,318649	0,0826025	0,917398	0,0446239	0,955376	0,946672	0,0533281	0,348332	0,651668	0,0317843	0,968216
0,0663183	0,933682	0,0990851	0,900915	0,340476	0,659524	0,024088	0,975912	0,226851	0,773149	0,109182	0,890818
0,964618	0,0353819	0,915881	0,0841193	0,913394	0,086606	0,861325	0,138675	0,578697	0,421303	0,729655	0,270345
0,0834728	0,916527	0,132581	0,867419	0,263468	0,736532	0,0148334	0,985167	0,803157	0,196843	0,392158	0,607842
0,974369	0,025631	0,923471	0,0765293	0,91912	0,0808798	0,870513	0,129487	0,449284	0,550716	0,757671	0,242329
0,204093	0,795906	0,0528667	0,947133	0,199697	0,800303	0,0298216	0,970178	0,285739	0,714261	0,0583147	0,941685
0,0612759	0,938724	0,0534813	0,946519	0,146869	0,853131	0,288703	0,711297	0,431843	0,568157	0,328416	0,671584
0,0522268	0,947773	0,0747591	0,925241	0,199599	0,800401	0,0176223	0,982378	0,194301	0,805699	0,0476413	0,952359
0,287658	0,712342	0,103089	0,896911	0,329324	0,670676	0,0296506	0,970349	0,428564	0,571436	0,0332855	0,966715
0,9681	0,0318998	0,94093	0,0590697	0,885347	0,114653	0,894448	0,105552	0,325139	0,674861	0,885518	0,114482
0,107015	0,892985	0,0667381	0,933262	0,15826	0,84174	0,0333693	0,966631	0,791954	0,208046	0,324976	0,675024
0,968589	0,0314108	0,928786	0,0712143	0,905166	0,0948337	0,866811	0,133189	0,441436	0,558564	0,724653	0,275347
0,973098	0,0269022	0,931142	0,0688576	0,913408	0,0865923	0,879399	0,120601	0,625691	0,374309	0,808066	0,191934
0,316409	0,683591	0,0972812	0,902719	0,0607856	0,939214	0,825236	0,174764	0,710109	0,289892	0,799396	0,200605
0,369651	0,630349	0,0974413	0,902559	0,120112	0,879888	0,0403891	0,959611	0,73967	0,26033	0,0524918	0,947508
0,108926	0,891074	0,0269342	0,973066	0,51414	0,48586	0,0573878	0,942612	0,597261	0,402739	0,226287	0,773713
0,969716	0,0302835	0,975616	0,0243843	0,160436	0,839564	0,969668	0,0303317	0,732936	0,267064	0,916561	0,0834386
0,0574262	0,942574	0,0424457	0,957554	0,250672	0,749328	0,014325	0,985675	0,304372	0,695628	0,132446	0,867554
0,980769	0,0192313	0,909376	0,0906238	0,908723	0,0912772	0,880892	0,119108	0,535386	0,464614	0,649237	0,350763
0,164125	0,835875	0,0704812	0,929519	0,145113	0,854887	0,189391	0,810608	0,785566	0,214434	0,0854075	0,914593
0,96848	0,0315199	0,9496	0,0504003	0,809048	0,190952	0,913502	0,0864976	0,32261	0,67739	0,926049	0,0739511
0,265717	0,734284	0,0862719	0,913728	0,0392949	0,960705	0,892535	0,107465	0,604447	0,395553	0,913233	0,0867674
0,0482668	0,951733	0,0725713	0,927429	0,117697	0,882303	0,0543262	0,945674	0,713149	0,286851	0,692502	0,307498

Оптимизация алгоритма

- Генетический алгоритм
- Метод Брента

Оптимизация алгоритма

- Параметры выделения импульсов акустической эмиссии:
  - порог импульса – 2;
  - длина интервала времени для определения среднего значения локальных максимумов – 5;
  - длина интервала времени для определения среднего абсолютного значения амплитуды сигнала – 200;
  - «отступы» по краям импульса – 18.
- Параметры вейвлет-преобразования:
  - количество шагов – 3;
  - фильтр – Добеши размера 8.
- Общие параметры алгоритмов сравнения:
  - степень для изменения расстояний – 2.
- Параметры мультиэвристического подхода:
  - способ сравнения сигналов – на основе коэффициента корреляции;
  - размер «окна» для заглядывания вперёд – 5;
  - размер массива для подходящих для сдвига позиций – 24;
  - «цена» сдвига сигнала на одну позицию – 0.46;
  - интервал между отсчётами (необходим для определения коэффициента корреляции) – 800;
  - стоимость «удаления» или «вставки» (при использовании специальной версии алгоритма вычисления расстояния Лёвенштейна для сравнения сигналов) – 0.46.

Оптимизация алгоритма

- Параметры функции корреляции:
  - интервал между отсчётами – 800.
- Параметры специальной версии алгоритма сравнения строк для сравнения сигналов:
  - штраф за пропуск одного из значений сигнала – 0,5.
  - интервал между отсчётами – 800;
- Параметры, связанный с характеристиками сигналов акустической эмиссии:
  - порог импульса – 1.3;
  - весовой коэффициент для числа импульсов АЭ, превышающих пороговый уровень, за единицу времени – 0.46.
  - весовой коэффициент для частоты импульсов АЭ, превышающих пороговый уровень, за единицу времени – 0.11.
  - весовой коэффициент для энергии импульсов АЭ, превышающих пороговый уровень, за единицу времени – 0.35.
  - весовой коэффициент для числа импульсов АЭ за единицу времени – 0.17.
  - весовой коэффициент для частоты импульсов АЭ за единицу времени – 0.22.
  - весовой коэффициент для энергии импульсов АЭ за единицу времени – 0.20.
- Параметры алгоритма кластеризации:
  - q – параметр кластеризации – 1.17.

Применение мульэвристического подхода к задаче сравнения ДНК

Математическая модель ДНК

- $$dna = (a_1, a_2, \dots, a_n),$$

$$n \in \mathbb{N}, a_i \in \{\mathrm{a}, \mathrm{c}, \mathrm{g}, \mathrm{t}\}.$$

Применение мультиэвристического подхода

Вход: Строки x и y.

Шаг 1: i := 0, j := 0, r := 0;

Шаг 2: if *x[i]* = *y[j]* then begin

сдвигаем обе строки;

r := *r* + стоимость совпадения символов *x[i]* и *y[j]*;

end

else begin

применяем эвристики для генерации возможных «траекторий» сдвига в позиции *i'* и *j'* таких, что *x[i']* = *y[j']*;

оцениваем их с помощью других эвристик;

усредняем полученные оценки, используя функцию риска;

осуществляем сдвиг (при этом может измениться значение *r*);

end;

Шаг 3: повторяем второй шаг до тех пор, пока не достигнут конец одной из строк.

Используемые эвристики

- Выбираем траектории, для которых выражение $(i' - i) + (j' - j)$ принимает минимальное, либо близкое к минимальному значение
- Сдвигаем ту строку, текущий символ которой реже встречается в другой строке.
- Комбинация двух предыдущих: результирующая оценка позиции складывается из её оценок первой и второй эвристиками.
- Используем алгоритм для поиска наибольшей общей подпоследовательности строк $x[i..i+k]$ и $y[j..j+k]$, где $k \sim 15$. Для сдвига выбираем такие индексы i', j', в которых заканчивается наибольшая общая подпоследовательность.

Используемые эвристики

- Комбинация третьей и четвёртой эвристик: оценка позиции складывается из её оценок обеими эвристиками. Оценка позиции (*i'*, *j'*) четвёртой эвристикой является отношением длины наибольшей общей подпоследовательности строк *x[i..i']* и *y[j..j']* к средней длине сдвига строк из позиции (*i*, *j*) в позицию (*i'*, *j'*).
- Используем алгоритм Нидлмана-Вунша для строк *x[i..i+k]* и *y[j..j+k]*, где *k* ~ 15. Сдвигаем строки в позицию (*i'*, *j'*), для которой соответствующее значение в таблице алгоритма Нидлмана-Вунша является наибольшим.
- Комбинация третьей и шестой эвристик: оценка позиции складывается из её оценок обеими эвристиками. Оценка позиции (*i'*, *j'*) шестой эвристикой является отношением значения в таблице алгоритма Нидлмана-Вунша, соответствующего этой позиции, к средней длине сдвига строк из позиции (*i*, *j*) в позицию (*i'*, *j'*).

Алгоритм Нидлмана-Вунша

$$D(i, j) = \begin{cases} d*i, \text{ если } j = 0 \\ d*j, \text{ если } i = 0 \\ max(D(i-1, j-1) + S(x_i, y_j), D(i-1, j) + d, D(i, j-1) + d), \end{cases}$$

Алгоритм Нидлмана-Вунша

	a	g	c	t
a	10	-1	-3	-4
g	-1	7	-5	-3
c	-3	-5	9	0
t	-4	-3	0	8

Алгоритм Нидлмана-Вунша

	g	g	a	t	t	a	c	c	t	t
t	0	-5	-10	-15	-20	-25	-30	-35	-40	-45
g	-5	7	2	-3	-8	-13	-18	-23	-28	-33
c	-10	2	4	2	-3	-8	-4	-9	-14	-19
t	-15	-3	-1	12	10	5	0	-4	-1	-6
c	-20	-8	-6	7	12	7	14	9	4	-1
a	-25	-13	2	2	7	22	17	12	7	2
c	-30	-18	-3	2	2	17	31	26	21	16
a	-35	-23	-8	-3	-2	12	26	28	23	18
c	-40	-28	-13	-8	-3	7	21	35	30	25
a	-45	-33	-18	-13	-8	7	16	30	31	26

Таблица сходства для Peponocephala electra (бесклювый дельфин)

Peponocephala electra									
Номер эвристики	1	2	3	4	5	6	7	8	9
Bison bison	0,55	0,40	0,58	0,58	-0,10	0,26	0,30	0,81	0,73
Bos taurus	0,54	0,40	0,58	0,58	-0,14	0,26	0,24	0,81	0,72
Canis lupus	0,55	0,41	0,68	0,60	-0,06	0,26	0,35	0,80	0,72
Drosophila simulans	0,51	0,37	0,55	0,56	-0,39	0,23	-0,24	0,59	0,40
Felis catus	0,56	0,41	0,58	0,57	-0,04	0,26	0,27	0,78	0,70
Gadus morhua	0,55	0,40	0,57	0,57	0,05	0,25	0,37	0,74	0,61
Gallus gallus	0,55	0,40	0,57	0,57	-0,05	0,25	0,25	0,71	0,55
Homo sapiens	0,55	0,40	0,57	0,57	-0,08	0,26	0,13	0,77	0,66
Mus musculus	0,55	0,41	0,67	0,58	-0,16	0,27	0,31	0,79	0,69
Orcaella brevirostris	0,57	0,78	0,91	0,94	0,40	0,34	0,85	0,94	0,93
Orcinus orca	0,58	0,64	0,87	0,91	0,46	0,37	0,87	0,93	0,93
Pan troglodytes	0,55	0,41	0,62	0,62	-0,06	0,27	0,29	0,79	0,69
Peponocephala electra	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Rattus norvegicus	0,55	0,40	0,63	0,59	-0,15	0,27	0,29	0,79	0,69
Sus scrofa taiwanensis	0,55	0,41	0,58	0,58	-0,16	0,27	0,28	0,78	0,67

Алгоритм поиска длины наибольшей общей подпоследовательности

Математическая постановка задачи

- $\mathrm{cs}(x,y) = w = (w_1, w_2, \dots, w_k), |x| = n, |y| = m, 0 \le k \le \min(n,m) \Leftrightarrow w = \varepsilon$ или $\exists\, i,j \in \mathbb{N}, i \le n, j \le m$: $w_k = x_i = y_j$,

  $x' = (x_1, x_2, \dots, x_{i-1}), y' = \left(y_1, y_2, \dots, y_{j-1}\right)$,

  $w' = (w_1, w_2, \dots, w_{k-1}), w' = \mathrm{cs}\left(x', y'\right)$

$$lcs(x,y) = \max_{|cs(x,y)|} cs(x,y)$$

Метод динамического программирования

		c	a	g	c	a	g	g	t	c	g	t	c	g	c	g	c
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
c	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
t	1	1	1	1	1	1	1	2	2	2	3	3	3	3	3	3
t	1	1	1	1	1	1	1	2	2	2	3	3	3	3	3	3
a	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
c	1	2	2	3	3	3	3	3	3	3	3	4	4	4	4	4
c	1	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5
c	1	2	2	3	3	3	3	3	4	4	4	5	5	5	5	6
a	1	2	2	3	4	4	4	4	4	4	4	5	5	5	5	6
a	1	2	2	3	4	4	4	4	4	4	4	5	5	5	5	6
c	1	2	2	3	4	4	4	4	5	5	5	5	5	6	6	6
a	1	2	2	3	4	4	4	4	5	5	5	5	5	6	6	6
a	1	2	2	3	4	4	4	4	5	5	5	5	5	6	6	6
t	1	2	2	3	4	4	4	5	5	5	6	6	6	6	6	6
t	1	2	2	3	4	4	4	5	5	5	6	6	6	6	6	6
c	1	2	2	3	4	4	4	5	6	6	6	7	7	7	7	7

O(n^2) по времени,

O(n^2) по памяти

$$a[i,j] = \begin{cases} 0, & \text{если } i == 0 \text{ или } j == 0 \\ a[i-1,j-1] + 1, & \text{если } x[i] == y[j] \\ \max(a[i-1,j], a[i,j-1]), & \text{если } x[i] \; != y[j] \end{cases}$$

Новый алгоритм

- изменение порядка перебора ячеек матрицы;
- использование массивов длин префиксов;
- использование списков вхождения символов.

Сложность по времени:

$$O(s \cdot (1 - r^2/n^2))$$

где s – количество пар (i,j), таких, что $x_i = y_j$, r – длина lcs.

Сложность по памяти:

$$O(n).$$

Новый алгоритм

	c	a	g	c	a	g	g	t	c	g	t	c	g	c	g	c
t	0	0	0	0	0	0	0	1	1							
c	1	1	1	1	1	1	1	1	2	2						
t	1	1	1	1	1	1	1	2	2	2	3					
t	1	1	1	1	1	1	1	2	2	2	3	3				
a	1	2	2	2	2	2	2	2	2	2	3	3	3			
c	1	2	2	3	3	3	3	3	3	3	3	4	4	4		
c	1	2	2	3	3	3	3	3	4	4	4	4	4	5	5	
c	1	2	2	3	3	3	3	3	4	4	4	5	5	5	5	6
a	1	2	2	3	4	4	4	4	4	4	4	5	5	5	5	6
a		2	2	3	4	4	4	4	4	4	4	5	5	5	5	6
c			2	3	4	4	4	4	5	5	5	5	5	6	6	6
a				3	4	4	4	4	5	5	5	5	5	6	6	6
a					4	4	4	4	5	5	5	5	5	6	6	6
t						4	4	5	5	5	6	6	6	6	6	6
t							4	5	5	5	6	6	6	6	6	6
c								5	6	6	6	7	7	7	7	7

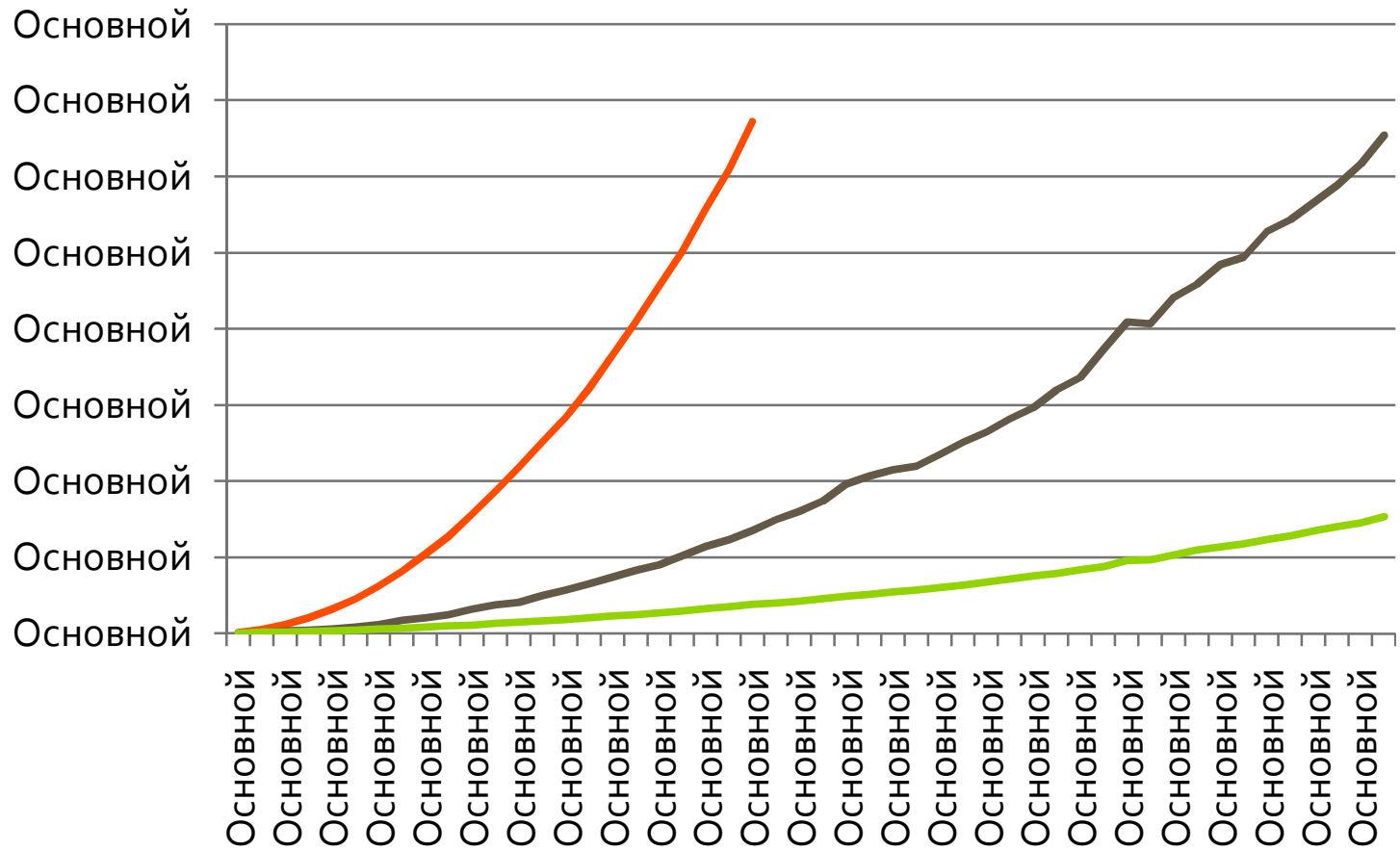
Новый алгоритм

	c	a	g	c	a	g	g	t	c	g	t	c	g	c	g	c
t								1								
c	1			1					2							
t								2			3					
t								2			3					
a		2			2											
c	1	2		3					3			4		4		
c	1			3					4			4		5		
c	1			3					4			5		5		6
a		2			4											
a		2			4											
c				3					5			5		6		6
a					4											
a					4											
t								5			6					
t								5			6					
c									6			7		7		7

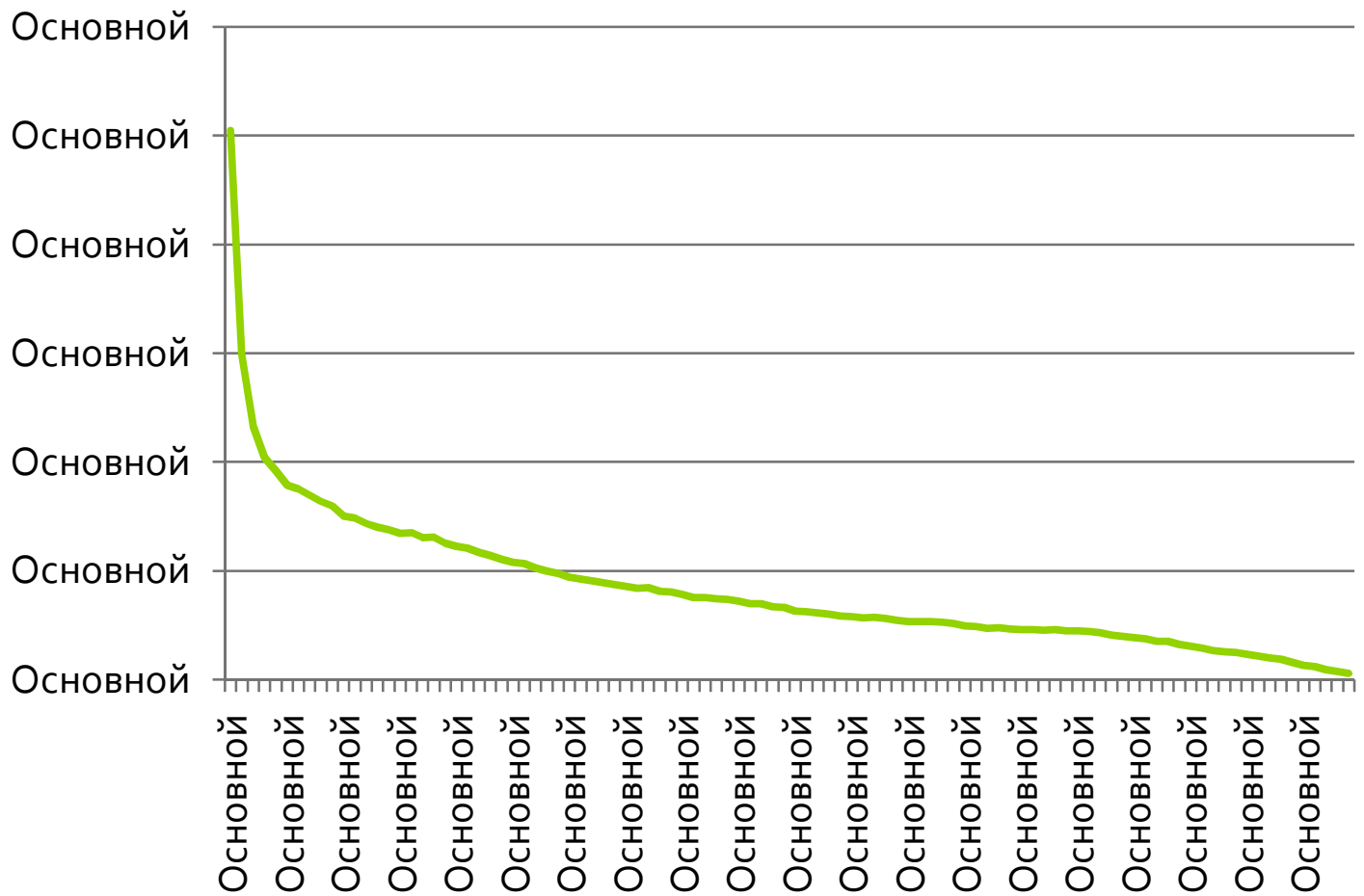
Алгоритм приближённого поиска

	c	a	g	c	a	g	g	t	c	g	t	c	g	c	g	c
t																
c	1			1												
t																
t																
a		2			2											
c				3												
c				3					4							
c									4							
a					4											
a																
c									5			5				
a																
a																
t											6					
t																
c												7		7		7

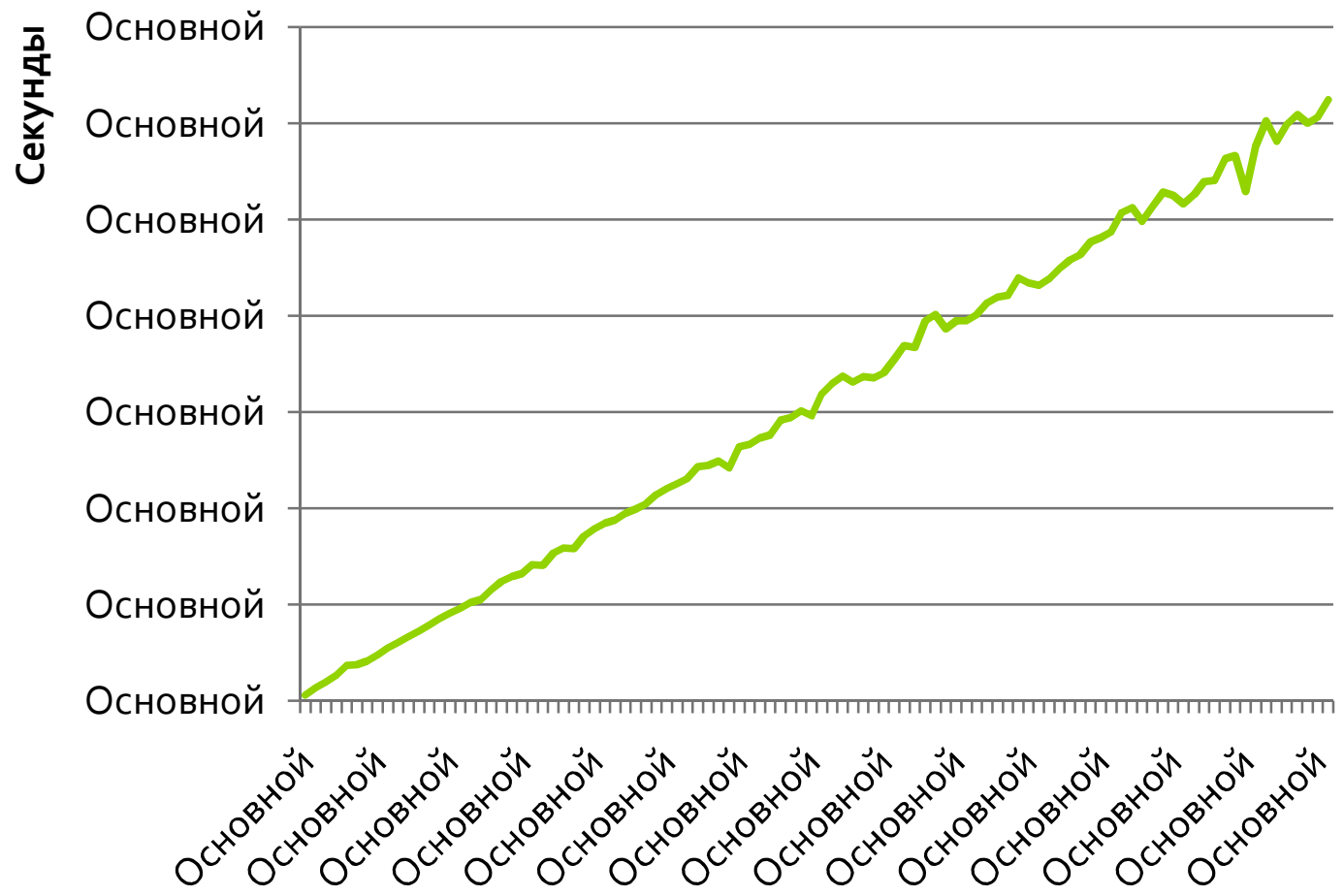
Точный алгоритм: производительность



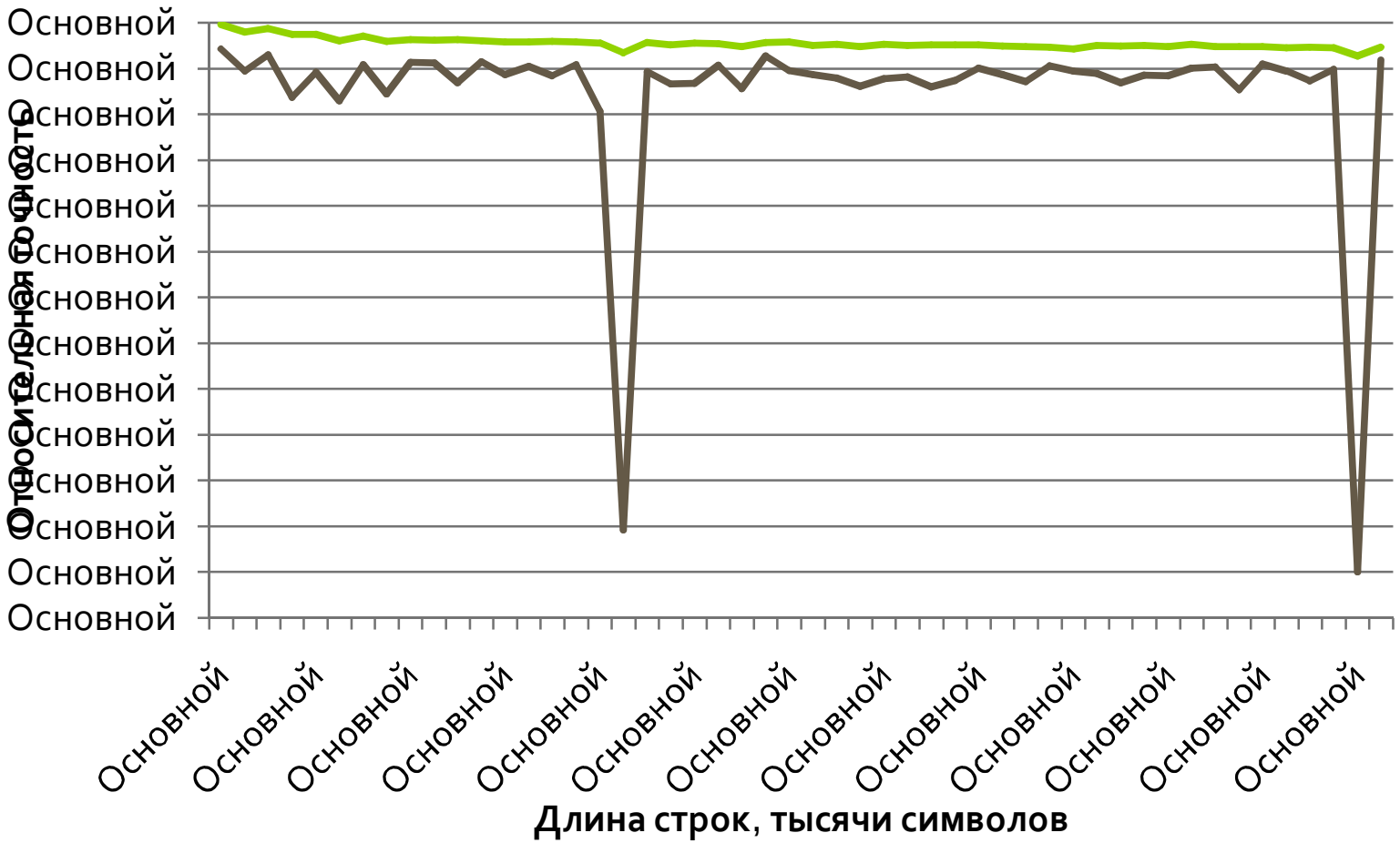
Точный алгоритм: погрешность прогноза результата



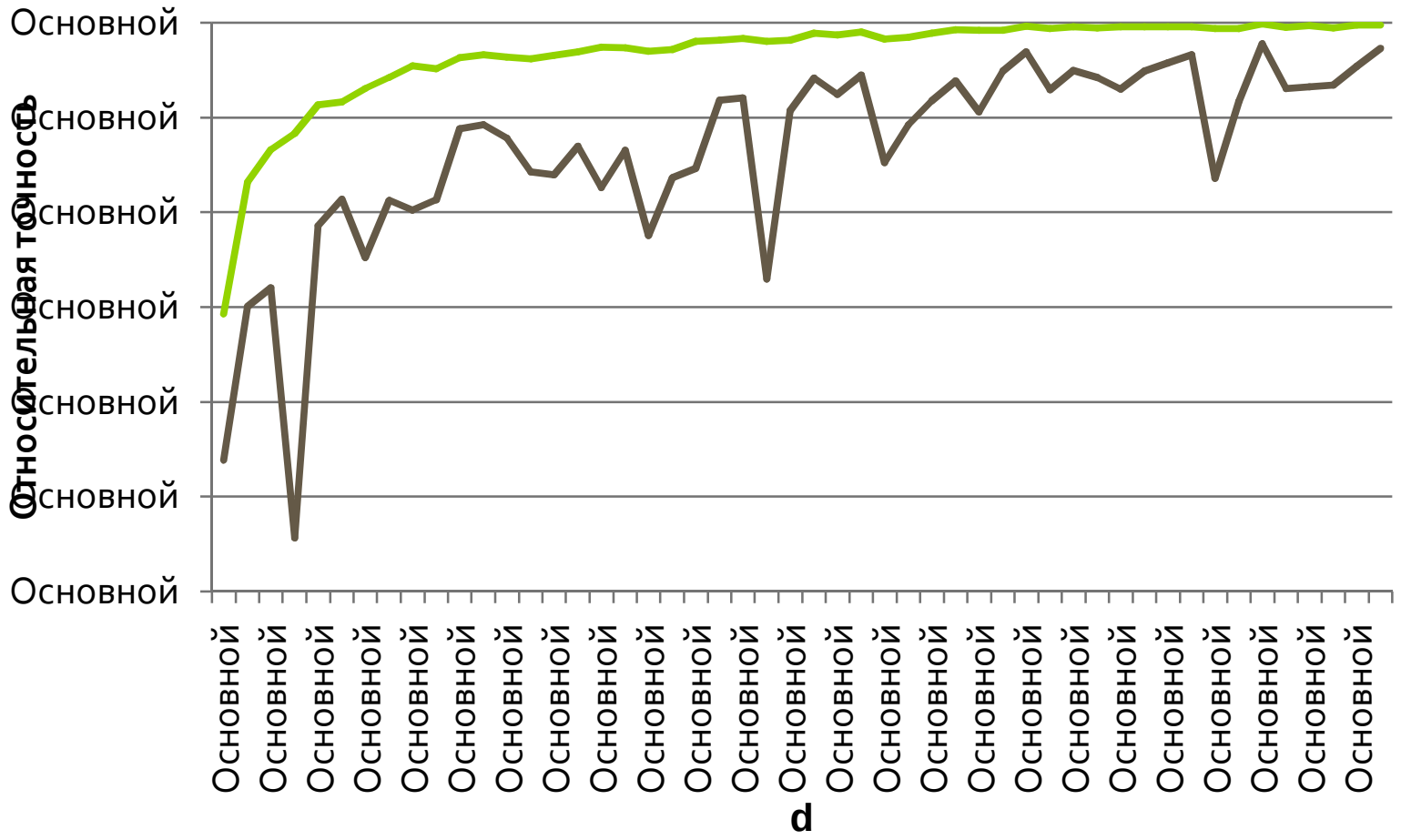
Неточный алгоритм: производительность



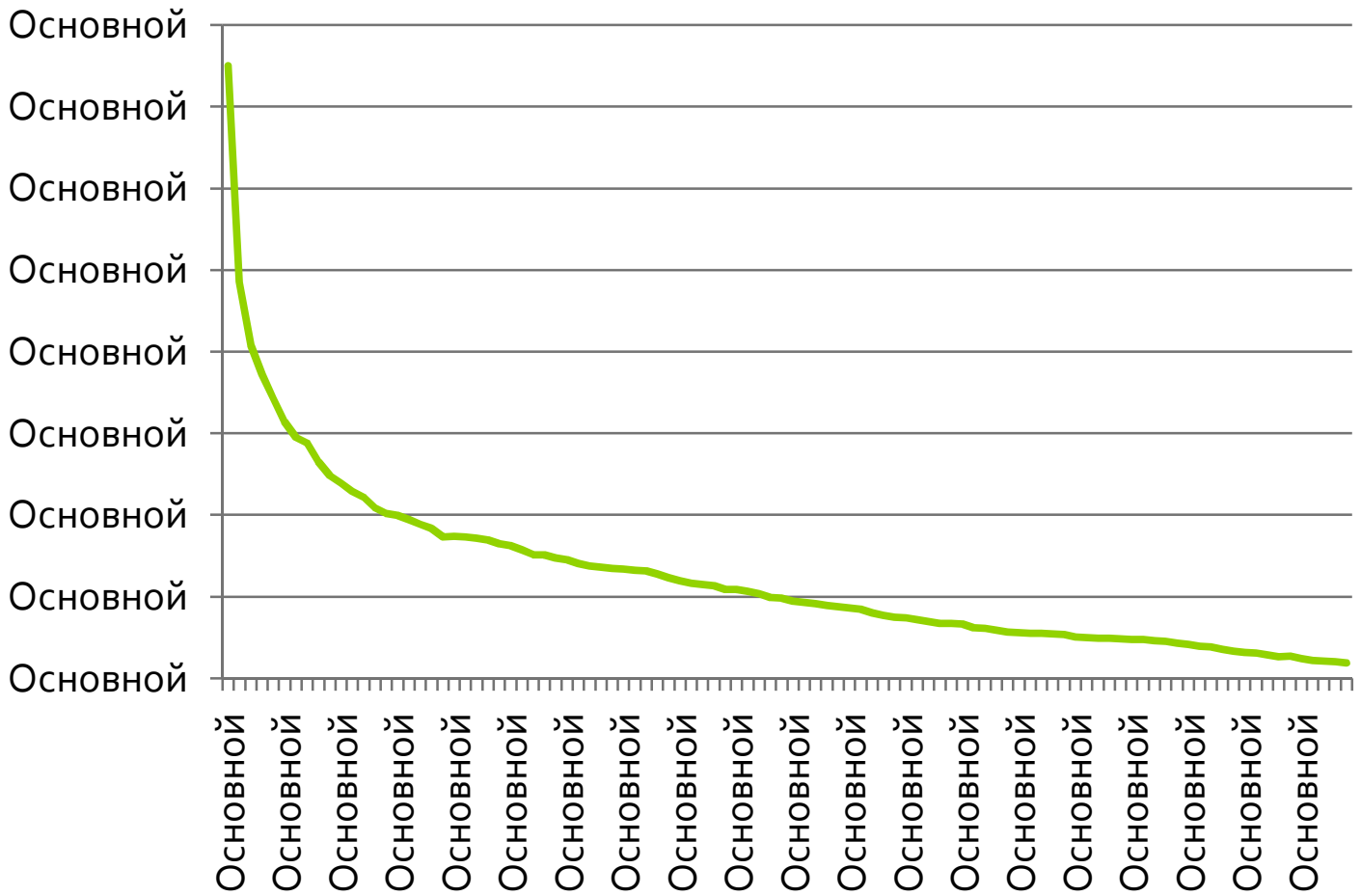
Неточный алгоритм: точность результата



Неточный алгоритм: точность в зависимости от глубины перебора



Неточный алгоритм: погрешность прогноза результата



Спасибо за внимание!