## Университет ИТМО

Факультет программной инженерии и компьютерной техники

## **Лабораторная работа №2** по "Алгоритмам и структурам данных" Тимус

Выполнил: Студент группы Р3206 Сорокин А.Н. Преподаватели: Косяков М.С. Тараканов Д.С.

## Задача №3 "Медиана на плоскости"

Решение за O(n).

Необходимо найти 2 точки такие, что между ними будут находится равное количество точек. Очевидно, что прямую, удовлетворяющую заданному условию можно провести ИЗ любой точки в какую-то конкретную точку. Таких прямых будет  $\frac{n}{2}$  из  $\frac{n(n-1)}{2}$  всех существующих прямых. Для удобства (и адекватных вычислений угла) мы берем крайнюю нижнюю (а из крайних нижних — самую левую) точку как начальную и ищем точку, в которую из начальной надо провести прямую.

Прошлое решение сортировало точки по полярному углу и выводило серединную точку. Однако, информация о положении других точек нас не интересует. Необходимо более быстрое решение, которое направлено на нахождение конкретной точки.

Уберем нулевую точку, получим массив нечетной длины (из условия становится очевидным) из разных точек (тоже очевидно, смотреть условие), и задача сводится к нахождению k-ой порядковой статистики.

Quickselect — алгоритм, разделяющий массив на 2 части — в одной находятся элементы, меньшие опорного, в другой — большие. Определяется опорный элемент pivot (каким-то образом, обычно берется первый/последний элемент) и далее за линию массив проходится до этого элемента, все меньшие элементы свапая (partition). Затем в зависимости от k выбираются элементы, слева или справа от pivot. Если k = uhdekcom pivot, то алгоритм останавливается.

Quickselect работает в среднем за линию.

Каждую итерацию от массива откусывается примерно половина, поэтому время можно рассчитать как:

$$T(n)=n+T(\frac{n}{2})=\sum_{i=0}^{\infty}\frac{n}{2^{i}}=2\,n=O(n)$$
? , по свойству УГП.

Однако, если на вход подается уже отсортированный массив, то Quickselect будет "откусывать" по 1 элементу каждую итерацию, что приведет к:

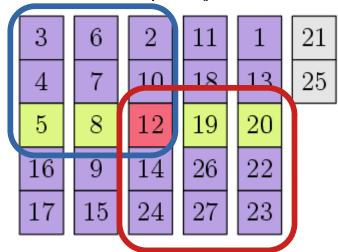
$$T(n)=n+T(n-1)=\sum_{i=0}^{n-1}(n-i)=\frac{n(n+1)}{2}=O(n^2)$$
 - Cymma A $\Pi$ .

Неудачный выбор опорного элемента может привести к увеличению времени до  $O(n^2)$ . Следовательно, необходимо на каждой итерации выбирать элемент удачно, чтобы можно было отделять как можно большие участки массива.

Решением данной проблемы является алгоритм **BFPRT** (или **Median of Medians**), позволяющий гарантированно находить удачный опорный элемент.

Массив разделяется на участки по 5 элементов, в каждом из которых находится медиана (например, сортировкой — поскольку размер участка фиксированно меньше либо равен 5, то время нахождения медианы для каждого O(1)). Затем, если кол-во медиан больше 1, медиана находится уже среди них. Таким образом, полученная медиана медиан является наиболее удачным опорным элементом.

Чтобы понять это, взглянем на картинку:



Здесь желтым обозначены медианы, а красным — медиана медиан. По картинке можно понять, что если индекс искомого элемента меньше индекса 12, то из массива гарантированно исчезает весь красный квадрат, а если больше — весь синий квадрат. Во входном массиве 27

элементов. Значит, такой выбор опорного элемента гарантированно отсекает чуть более 30% элементов на каждой итерации.

Почему это работает за линию:

Медиана на каждом участке ищется за константу, участков  $\frac{n}{5}$ . Если медиан больше 1, поиск производится рекурсивно -  $T(\frac{n}{5}) = T(\frac{2n}{10})$ . Поскольку (гарантированно) удачный выбор опорного элемента уменьшает вектор в худшем случае на 30%, то  $T(\frac{7n}{10})$ . Так же некоторые операции происходят за линию (например, partition) - Cn. Тогда имеем:

$$T(n) = T\left(\frac{2n}{10}\right) + T\left(\frac{7n}{10}\right) + Cn = Cn + \left(\frac{2}{10} + \frac{7}{10}\right)Cn + \left(\frac{2}{10} + \frac{7}{10}\right)\left(\frac{2}{10} + \frac{7}{10}\right)Cn + \dots = Cn * \sum_{i=0}^{\infty} \left(\frac{9}{10}\right)^{i} = 10 Cn = O(n)$$

```
Код решения:
#include <algorithm>
#include <cmath>
#include <iostream>
#include <utility>
#include <vector>
using namespace std;
struct Point {
 long long x, y, i;
 double a = 0;
};
bool compare(const Point a, const Point b) { return a.a < b.a; }
Point findKMin(vector<Point> v, int l, int r, int k) {
 // BFPRT
 int p1 = 1, p2 = r;
 while (p1 < p2) {
  if (p1 \ge p2) {
   break;
  int n = p2 - p1 + 1;
```

```
if (n < 5) {
    sort(v.begin() + p1, v.begin() + p1 + n, compare);
    int m = (p1 * 2 + n) / 2;
    swap(v[p1], v[m]);
    break;
   }
  int p = p1;
  for (int i = p1; i < n / 5; ++i) {
    int p_{left} = p1 + i * 5;
    int p_right = p1 + i * 5 + 5;
    sort(v.begin() + p_left, v.begin() + p_right, compare);
    int m = (p_left + p_right) / 2;
    swap(v[p], v[m]);
    ++p;
  }
  p2 = p - 1;
 // partition
 int i = r;
 for (int j = r; j \ge 1; --j) {
  if (v[j].a < v[l].a) {
    swap(v[j], v[i]);
   --i;
 swap(v[l], v[i]);
 if (i == k) {
  return v[i];
 \} else if (i > k) {
  return findKMin(v, l, i - 1, k);
 return findKMin(v, i + 1, r, k);
}
int main() {
```

```
int n;
cin >> n;
long long x, y;
vector<Point> vp;
Point downmostPoint = {(long long)10e7, (long long)10e7, 0};
for (int i = 1; i \le n; ++i) {
 cin >> x >> y;
 vp.push_back({x, y, i});
 if (y < downmostPoint.y) {
  downmostPoint = \{x, y, i\};
 } else if (y == downmostPoint.y) {
  if (x < downmostPoint.x) {
   downmostPoint = \{x, y, i\};
  }
 }
swap(vp.back(), vp[downmostPoint.i - 1]);
vp.pop_back();
for (int i = 0; i < vp.size(); ++i) {
 vp[i].x -= downmostPoint.x;
 vp[i].y -= downmostPoint.y;
 vp[i].a = atan2(vp[i].y, vp[i].x);
Point nextPoint = findKMin(vp, 0, vp.size() - 1, vp.size() / 2);
cout << downmostPoint.i << " " << nextPoint.i;</pre>
return 0;
```



Последние попытк

						Army princip, holy, 1000 - Name Brown, Street,			
10	Ann	Army	Jagens	Fran	Результат приверши	10	Herman padiress	Bappeners	
3002202	84 40 80 Evep 2024	phosp. Note. POR	270. Siverenung overnig	Chrys a 17 old	Accepted		0.046	7 18698	
10022021	21 dn m 7 map 2024	photo. Notic. 2024	27C.Biverorusarerus	Charges 27 old	Accepted		0.000	7 650 83.	
1912/09	21 dt 22 7 map 2024	photo. No. 60 (2004)	12°C. Mineronana amenina	Charges 27 old	Wang souser	1	0.001	624 KB	
100 <u>11</u> 002	29-21-26 7-map 2024	photo Newton PUID	1270. Mineronancements	Charge a 37 old.	Wang souser	4	0.000	652 KB	
10023004	20-20-20 7 map 2024	physical Physics (PDD)	27.8verouserers	Chergos 27 eld	Wang research	4	0.004	628 KB	
10021000	20-23-07 7 map 2324	physics 2004	1270. Mineron proprietores	Chrys a 27 old	Wang souser	4	DEED	624 KB	
10022000	20-22-48 7 mm 2024	private Novice (NDS)	1270, Moreoverserverse	Day 13.7 eld	Wang souser	4	DEED	620 KB	
lost line	29-22-22 7 map 2024	photo Note 200	27LSverousseres	Vessel Co. 2002 et 8	Wang source	4	DEED	102 KB	
Tool Taxe	20-20-23 7 mp 2024	physic North WD41	270 Moreovanamenta	Chrys. IT old	Wang source	4	0.001	636.95	
programs.	294349	prioriti. Protes PRINT	27LSverousserers	Chengo a 37 old	Warranner	1	0.001	652 KB	
2742765	7 map 2024 20:11:48	2000 Peter 200	27.2000000000	0137.66	Wareanne	1	0.001	620.60	
	7 map 2024 20-00-44			Day 13.7 old	Wang source	1	0.001	620.00	
20022003	7 map 2024 20,20,46	phinal Police PDR	290. Sween and over 12			,	DEST	600 KB	
See 17 cm	7 map 2024 20,000,02	photo.hoto.HDB	27LEveronsorers	0 13.7 e64	Wang source Suring your lawner				
Seeffeet.	7 map 2024	144-25,740-20-20-20-20-20-20-20-20-20-20-20-20-20	27LSverouserers	0 13.7 eld	Rantine enter (access stabilité)	2	0.001	620 KB	
Southern	20.20.59 7 mm 2026	H1045,71040,70287	27LSverouserers	Gas 13.7 e64	Weng souser	.7	0.835	680 KB	
Seefend	23:01:02 3 map 2024	Head From PDB	27LSverousserva	Gas 13.7 eld	Memory limit exceeded	6.	0.062	77.784 KB	
property	27.50 di 3 map 2024	PH-05.7000-700F	B7LSverouserers	Gas 13.7 old	Accepted		0.046	10.6 KB	
Seedles 673	22 5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	physik (1946) 2024	1270. Mineron proprieta	Day 13.7 old	Wasting over (person statem)	1	0.000		
property	22.20.28 3 map 2024	photo. No. 60 (1924)	27LBvernaumens	Sec 13.7 etd.	Wantires recor (possess statemen)	1	0.004		
(property	22 23 28 5 map 2324	phints Novice 2004	1270. Sweep runsprense	Sec 13.7 e64	Martine serie (process sinhelise)	8	0.001	301 KG	
pequan	22 GH 56 5 may 2024	PROBEED TO BE	270. Biverorusarens	Day 13.7 eM	Memory limit recorded		0.018	07480 88	
pequan	22 00 m	photo Screen PDS	27LSverouserers	Day 13.7 eld	Photiera ever (sinds	8	0.046	65 428 KB	
property	27 6 7 62 5 may 2024	photo hoto 200	1277, Monagon and agreement	Day 13.7 eld	Photies were (sinch meeting)		0.046	0.400	
leafeas?	21.0041	phone Protection	270. Moreovana america	0133.66	Warm system		0.004	300 KG	
property.	20:20:08	rend hour MDF	27.2000000000	013.7.66	Wara sone	26	0.033	ARC 103	
Toronto I	3 may 2024 23-80-50	Med Set SDS	27.200000000	0133 e64	Accepted	-	0.046	600 KB	
Toronton.	38 50 43			0137.66	Accepted		Date	600 KG	
	4 map 2024 14-00-01	photo.Note.PDI	27L2verousseres						
Socialist	4 map 2024 23 (20 27	photo.hoto.HDB	27LSverousserrs	0 13.7 eld	Accepted		0.046	680 KB	
Someone	4 may 2024 2040-07	Minds Protes PDB	27LSverouserers	0 13.7 eld	Accepted		0.046	684 KB	
Towns 177	35 sep 2006	photo.Neto.7008	27LSverouserers	G., 133 eM	Accepted		0.001	104 KB	
Sounded	30-510-30 35 seep 2004	photal Prote (RD4)	27LBveroussrens	Day 13.7 eld	Accepted		0.846	20.0 KB	
posição	29:50:59 35 sep 2006	photol North 2008	127LSverousserers	San 13.7 old	Weng souser	11	0.001	384 KB	
Secordar	39-51-61 35 way 2006	phints. Note: 2008	127LSverousserers	San 13.7 old	Weng souser	11	0.000	300 KB	
20100 <u>40</u>	33 sep 2004	phosp. Note. POR	270. Siverorus arrents	0132+64	Wang souser	1	0.001	300 KB	
possuja	20-27-32 35 sep-2006	physical Production	27LSverousserva	Gas 13.7 eld	Completion new				
possujas	20-20-61 35 sep 2006	physic Policies	270.8weensuranens	Chergo a 27 old.	Complete over				
30400 <u>400</u>	20-13-05 35 way 2006	physic Period PDIP	270. Niverona communica	Chengo a 37 eld.	Wang source	1	0.004	30.00	
104004_01	20-00000 35 seep 2006	physic Polic PDR	270. Mineronal communication	Chergo a 27 old	Wang sousse	1	0.001	300 (0)	
posseger.	20-00-00 25-000-00	PARESTON TO BE	B1LBverouserers	Chrys a 17 old	Wang souser	1	0.000	300.00	
possenge.	20.0057	phone State State	29. Sween and over 12	Charges 27 old	Wang source	1	0.001	300 (0)	
20100300	35 sep 2006 31 21 23 23	PROBLEMS TO STATE	27LEveryours	Dem. II dil	Wareanne	2	0.033	364.65	
10406302	35 sep 2006 39.01-68	Med Sub SDS		Divinia 27 et l	Ware some	1	0.004	30.0	
	35 sep 2006		27C. Sween and over 12						
Secured	35 sep 2006 85 06 06	phinal Police PDIP	290.26 veronum amenda	0133 eM	Worse source	4	0.001	A00 KG	
Secured	35 sep 2006	photo. Note: PDB	27LEverouserers	U 13.7 eld	Worspanner	4	0.011	380 KB	
To con serv	25 may 2004 20 mm	PH-25.7606-708	B1LB/reconstruct	Gas 13.7 eld	Wang assault	4	0.004	30 (0)	
Total Tata	27 sep 2006	photo.Note.RDB	1270. Minerona amenia	Gr: 133 eld	Worsp sousser	10	DEIL	.000 KB	
Seeding.	20:21:22 28 sep 200	physic 7000 7000	27LSverouserers	Gr-137 eld	Weng souser	10	0.000	.004 KB	
1000000	31.3134 31 sep 2016	photol. Notice PDM	ISON Proposition of the Control of t	0 to 133 eld	Wang souser	11	0.004	.000 KGs	
progress	31/27/52 31 way 2004	physic Protection	270. Mineronaus america	0 to 13.7 eld	Wang souser	11	0.000	300 KG	
pranage	20 01 05 28 sep 2006	photo Newton PUBL	27.Sverouserers	Gas 13.7 eld	Wang souser	2	0.001	381 (0)	
	10.130.1308		15120130						

## Задача №4 "В Стране Дураков"

Решение за O(n log n).

Задача сводится к построению последовательности такой, что никакие два знака одинакового типа не стоят рядом (если это возможно). Рассмотрим случаи, когда это невозможно:

- если *k*=1
- если  $\left|\left(\sum_{i=1}^k n_i 2*max(\{n_i\}_{i=1}^k)\right)\right| > 1$ . Чтобы понять данную запись, рассмотрим набор  $\{1, 2, 3\} = > \{4, 2, 2\}$ . Поскольку 8-2\*4=0, их можно расставить как  $1\ 2\ 1\ 3\ 1\ 2\ 1\ 3$ . Если рассмотреть  $\{1, 2, 3\} = > \{5, 2, 1\}$ , то 8-2\*5=-2 и  $1\ 2\ 1\ 3\ 1\ 1$ .

Как следует расставлять числа.

Очевидно, что массив необходимо отсортировать в порядке убывания и рассматривать максимальные элементы. Пусть  $\{n_1,n_2,...n_k\}$  , где  $n_1 \ge n_2 \ge ... \ge n_k$  . Тогда необходимо начинать формировать последовательность, чередуя  $a_1$  и  $a_2$  , затем, когда закончится  $a_2$  , выбрать  $a_3$  , затем выбрать  $a_4$  и так далее, чередуя элементы только 2x типов.

Однако данный подход не работает.

Рассмотрим пример:

3

354

После сортировки имеем:  $\{\{2:5\}, \{3:4\}, \{1:3\}\}$ 

Чередуем первые 2 элемента

23232323

Получаем  $\{\{2:1\}, \{3:0\}, \{1:3\}\}$ 

Выбираем последний элемент

2323232321

Остается  $\{\{2:0\}, \{3:0\}, \{1:2\}\}$ , и 2 элемента, которые некуда деть.

Данный подход провалился, поскольку в конце реализации всегда будет оставаться  $a_k \ge 1$  элемент. Это вызвано тем, что количество знаков разных типо уменьшается неравномерно, т.е на какой-то из итераций  $a_i \gg a_{i-2} > a_{i-1}$ .

Для решения данной задачи необходимо уменьшать количество знаков разных типов равномерно, то есть на каждой итерации уменьшать на 1 конкретно 2 типа с максимальным количеством знаков (первый и второй максимумы). Для этого на каждой итерации необходимо определять новые 1 и 2 максимумы.

```
Продемонстрирую алгоритм решения на прошлом примере:
3
354
После сортировки имеем: \{\{2:5\}, \{3:4\}, \{1:3\}\}
currentFirstMax = {2:5}, currentSecondMax = {3:4}
23
currentFirstMax = \{2:4\}, currentSecondMax = \{1:3\}
2321
currentFirstMax = {2 : 3}, currentSecondMax = {3 : 3}
232123
currentFirstMax = {1:2}, currentSecondMax = {2:2}
23212312
currentFirstMax = {3 : 2}, currentSecondMax = {1 : 1}
2321231231
currentFirstMax = \{2:1\}, currentSecondMax = \{3:1\}
232123123123
Получаем \{\{2:0\}, \{3:0\}, \{1:0\}\}
```

Теперь встает задача быстрого поиска нового максимума. Поскольку вызывать sort() в векторе на каждой итерации слишком затратно, необходимо использовать другую структуру. priority\_queue отлично справляется с задачей: из-за особенностей реализации очередь предоставляет поиск максимума за O(1) (максимум является корнем дерева), и вставку/удаление элементов за O(log n) (бинарный поиск в дереве).

```
Код решения:

#include <iostream>

#include <queue>

#include <string>

#define P pair<int, int>
```

```
using namespace std;
class Compare {
public:
 bool operator()(P a, P b) {
  if (a.second == b.second) {
   return a.first > b.first;
  return a.second < b.second;
 }
};
int main() {
 int k, n;
 cin >> k:
 priority_queue<P, vector<P>, Compare> pq;
 for (int i = 1; i \le k; ++i) {
  cin >> n;
  pq.push({i, n});
 string res = "";
 while (!pq.empty()) {
  P currentFirstMax = pq.top();
  pq.pop();
  res += to_string(currentFirstMax.first) + " ";
  --currentFirstMax.second;
  if (!pq.empty()) {
   P currentSecondMax = pq.top();
   pq.pop();
   res += to_string(currentSecondMax.first) + " ";
   --currentSecondMax.second;
   if (currentSecondMax.second > 0)
     pq.push(currentSecondMax);
  }
  if (currentFirstMax.second > 0)
   pq.push(currentFirstMax);
```

```
}
cout << res;
return 0;
}</pre>
```