

Dia 0 - Treinamento para maratona de programação

Emanuel Huber - emanuel.tesv@gmail.com



Planejamento

- Aula 0 - Introdução / Revisão: aquecimento, notação Big O
- Aula 1 - Estruturas: queue, stack, set, map, priority_queue
- Aula 2 - Paradigmas: complete search, divide and conquer, greedy, dynamic programming
- Aula 3 - String: matching, DP, suffix tree
- Aula 4 - Grafos: representações, DFS, BFS, Dijkstra, Kruskal, Edmonds-Karp dos números
- Aula 5 - Matemática: BigInteger, combinatória, teoria
- Aula 6 - Geometria computacional
- Aula 7 - Competição

Revisão C++

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    return 0;
```

```
}
```

Utilizando cin e cout

```
int number;
string str;
// Utilizando cin/cout
cout << "Insira um número inteiro: ";
cin >> number;
cout << "Número digitado: " << number << endl;
cout << "Insira um texto: ";
cin.ignore(); // Importante!
getline(cin, str);
cout << "String digitada: " << str << endl;
```

```
// Utilizando scanf/printf
printf("Insira um número inteiro: ");
scanf(" %d", &number);
printf("Número digitado: %d\n", number);

char str2[100];
printf("Insira outro texto: ");
scanf(" %[^\n]", str2);
printf("Nova string digitada: %s\n", str2);
```

Operadores binários

- AND:

```
int n1 = 1, n2 = 2;  
cout << (n1 & n2);
```

- Complement/NOT:

```
unsigned long max = 0;  
cout << ~max;
```

- Shift left:

```
int n1 = 4;  
cout << (n1 << 1);
```

- OR:

```
int n1 = 1, n2 = 2;  
cout << (n1 | n2);
```

- XOR:

```
int n1 = 2, n2 = 2;  
cout << (n1 ^ n2);
```

- Shift right:

```
int n1 = 16;  
cout << (n1 >> 1);
```

Complexidade de algoritmos

- Análise de tempo/memória que o algoritmo demora/utiliza para executar
 - Número de operações (10^8 operações \sim 1 segundo para executar)
- Possui várias notações Big O, Big Theta etc
- Notação Big O
 - Avalia o algoritmo quanto ao pior caso
 - $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$
- Leia mais em: <http://www.codcad.com/lesson/16>

Dada uma lista de números inteiros ordenados de tamanho N, procure um par de números cuja soma seja K

$N \leq 10^6$

```
int max = 1000000, list[max], n, k, i, j;
cin >> n >> k;
for (int i = 0; i < n; ++i)
    cin >> list[i];
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
        if (list[i] + list[j] == k) {
            cout<< i << " " << j <<endl;
            i = j = n;
        }
    }
}
```

Dada uma lista de números inteiros ordenados de tamanho N, procure um par de números cuja soma seja K

$N \leq 10^6$

```
int max = 1000000, list[max], n, k, i, j;
cin >> n >> k;
for (int i = 0; i < n; ++i)
    cin >> list[i];
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
        if (list[i] + list[j] == k) {
            cout<< i << " " << j <<endl;
            i = j = n;
        }
    }
}
```

$O(n^2)$

Vamos tentar melhorar!

Dada uma lista de números inteiros ordenados de tamanho N, procure um par de números cuja soma seja K

$N \leq 10^6$

```
int max = 1000000, list[max], n, k;
cin >> n >> k;
int i = 0, j = n - 1;
for (int i = 0; i < n; ++i)
    cin >> list[i];
while (i < j) {
    int soma = list[i] + list[j];
    if (soma < k)
        i++;
    else if (soma > k)
        j--;
    else {
        cout << i << " " << j << endl;
        break;
    }
}
```

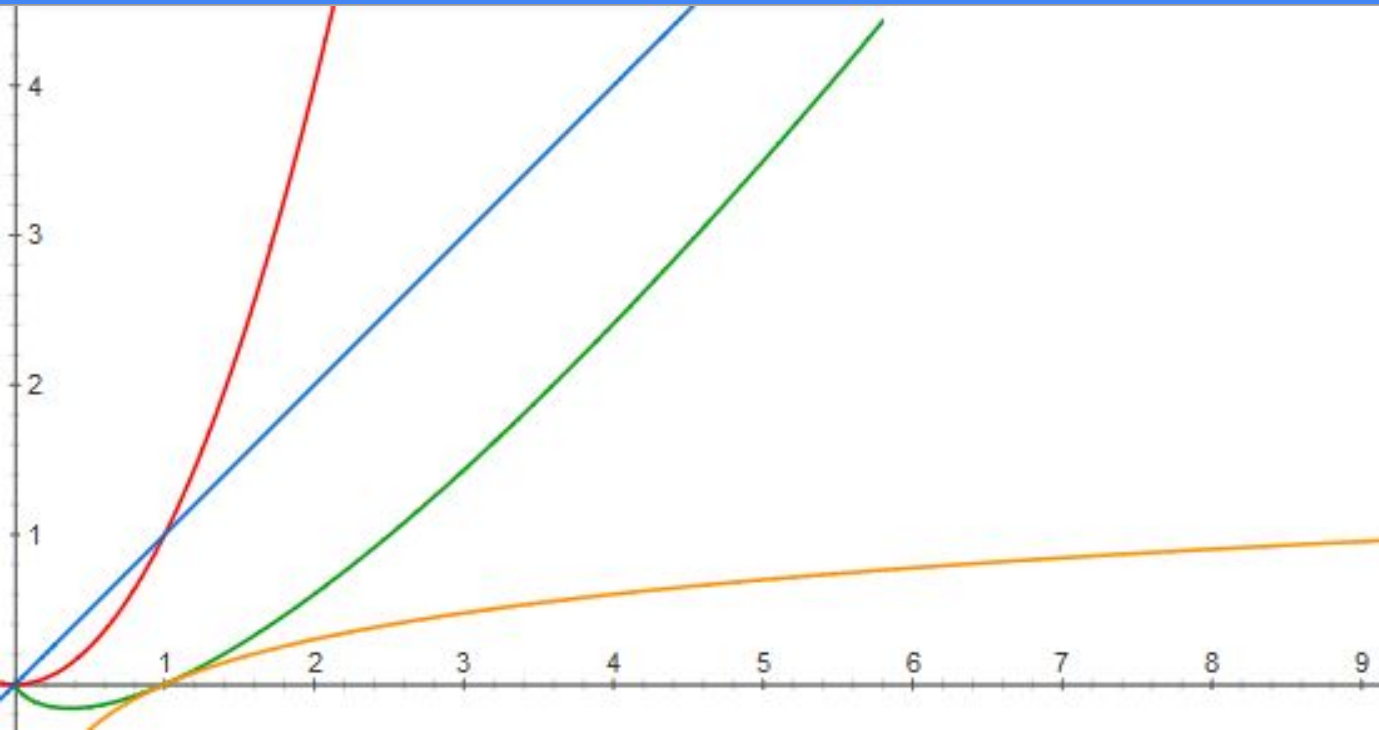
Dada uma lista de números inteiros ordenados de tamanho N, procure um par de números cuja soma seja K

$N \leq 10^6$

```
int max = 1000000, list[max], n, k;
cin >> n >> k;
int i = 0, j = n - 1;
for (int i = 0; i < n; ++i)
    cin >> list[i];
while (i < j) {
    int soma = list[i] + list[j];
    if (soma < k)
        i++;
    else if (soma > k)
        j--;
    else {
        cout << i << " " << j << endl;
        break;
    }
}
```

O(n)
Muito melhor!

Comportamento gráfico



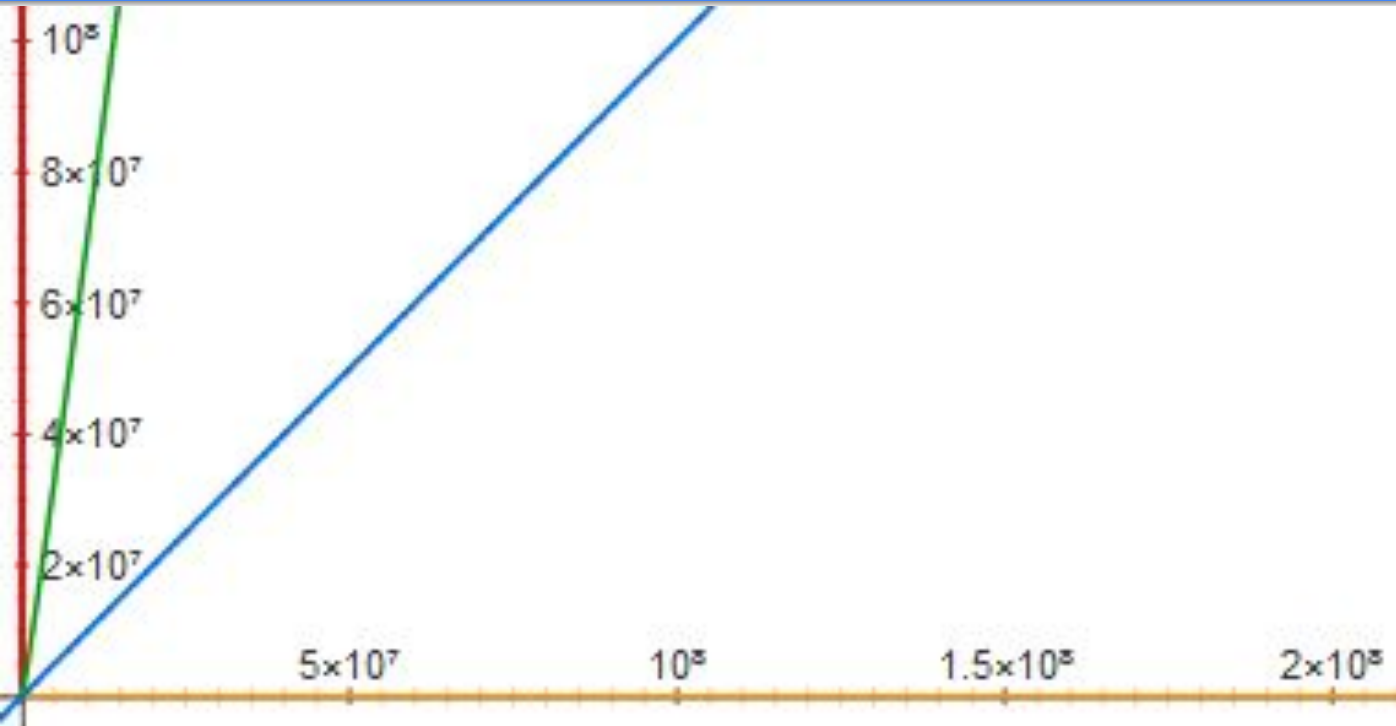
$O(\log n)$

$O(n)$

$O(n \log n)$

$O(n^2)$

Comportamento gráfico



$O(\log n)$

$O(n)$

$O(n \log n)$

$O(n^2)$

Exercícios

Uri online judge: <https://www.urionlinejudge.com.br/judge/en/tournaments/rank/670>