Maratona de Programação - Mini Curso

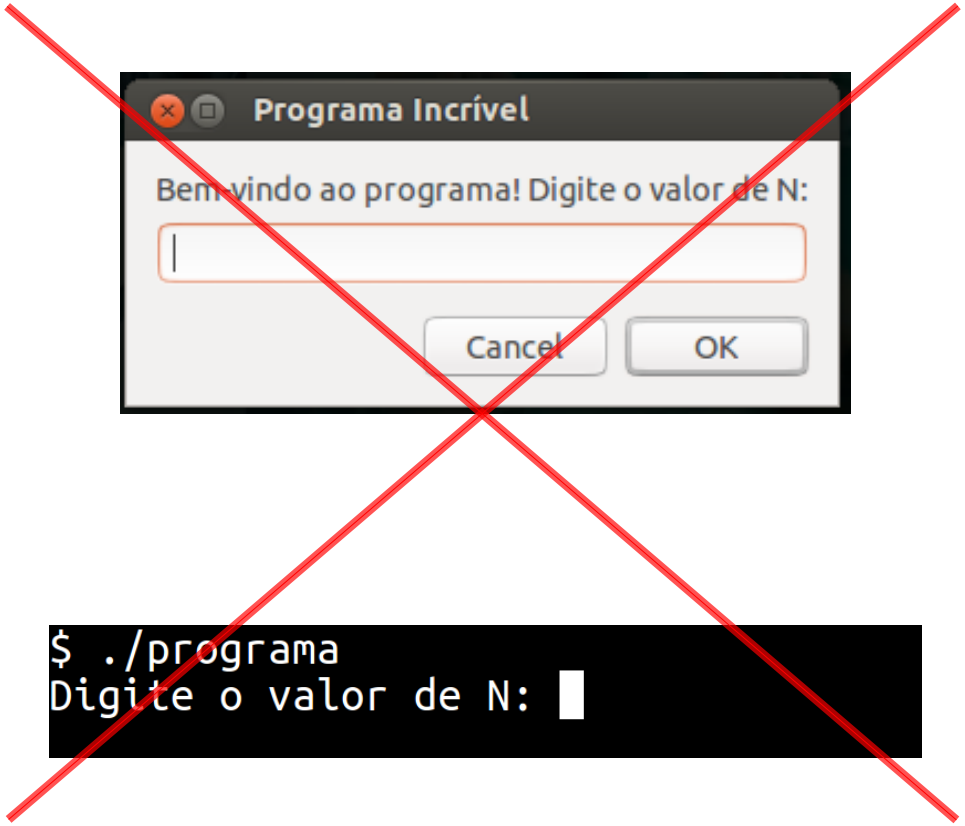
MEDITEC UTFPR

22 de Maio de 2013

Parte 1 - Como fazer

Parte 2 - O que fazer

Leitura




```
scanf("%d",&N);
printf("%d\n",N*N);
```

Vários casos de teste

- Até $N = 0$

```
scanf("%d",&n);
while (n != 0) {
    ...
    scanf("%d",&n);
}

while (scanf("%d",&n) and n) {
    ...
}
```

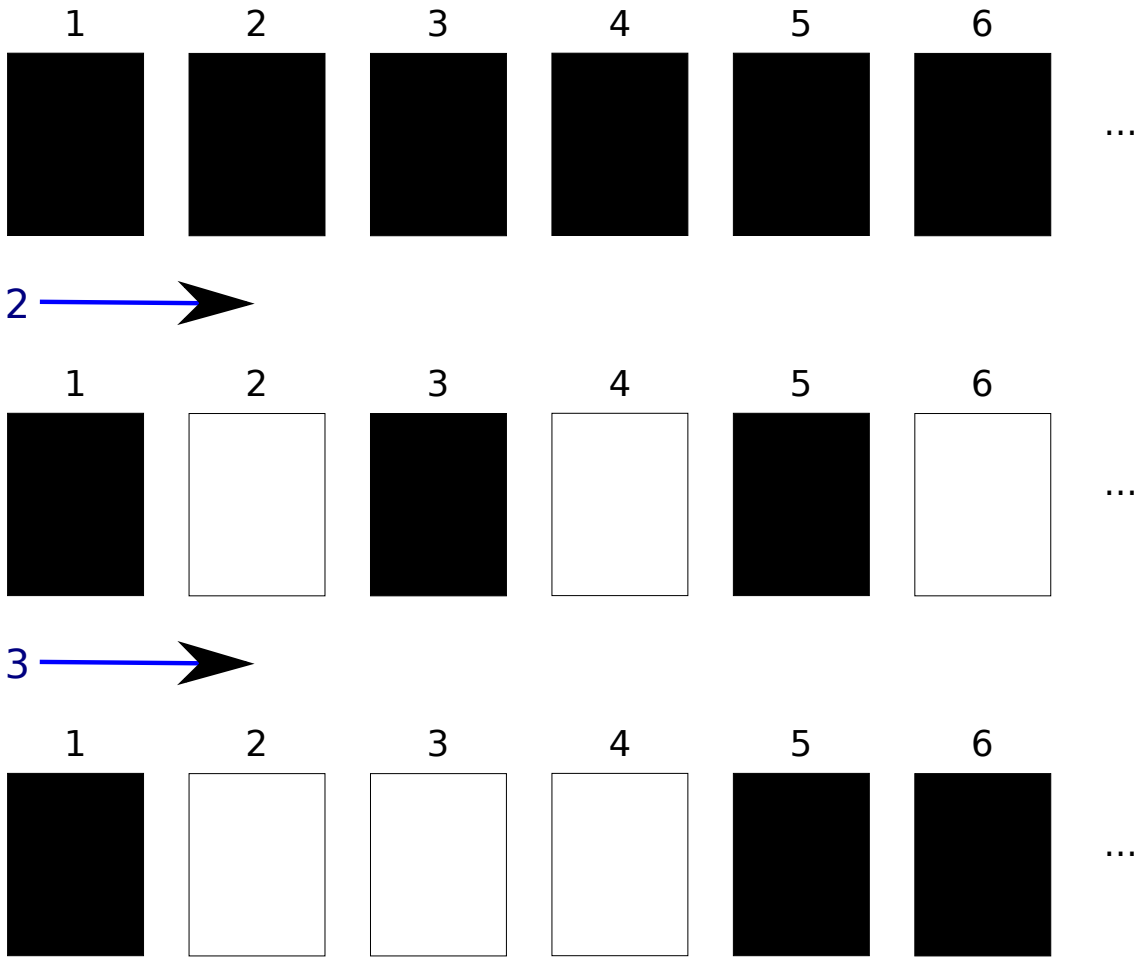
- Até EOF

```
while (scanf("%d",&n) != EOF) {
    ...
}
```

Complexidade esperada

- Linear em até $2 \times 10^7 = 2^{24}$;

N	Complexidade
10^{18}	$O(logN)$
10^9	$O(\sqrt{N})$
10^7	$O(N)$
10^6	$O(NlogN)$
10^4	$O(N\sqrt{N})$
1000	$O(N^2)$
100	$O(N^3)$
20	$O(2^N)$
10	$O(N!)$


1 2 3 4 5 6
...
2
1 2 3 4 5 6
...
3
1 2 3 4 5 6
...

Solução 1

```
Para i de 1 a N:
    estado[i] = 0;

Para i de 1 a N:
    Para j de i a N, de i em i:
        estado[j] = 1 - estado[j];

Para i de 1 a N:
    Se estado[i] = 1 então
        Imprime i;
```

Observação importante!

D(4): {1, 2, 4}

D(6): {1, 2, 3, 6}

D(12): {1, 2, 3, 4, 6, 12}

D(25): {1, 5, 25}

Solução 2

```
Para i de 1 a N:
    Se NumeroDivisores(i) é impar então
        Imprime i;
```

Outra Observação importante!

Se i divide N, então $\frac{N}{i}$ também divide!

D(12): {1, 2, 3, 4, 6, 12}

- $12 = \frac{12}{1}$
- $6 = \frac{12}{2}$
- $4 = \frac{12}{3}$
- $2 = \frac{12}{6}$
- $1 = \frac{12}{12}$

Impar se há um inteiro i tal que
$i = \frac{N}{i} \therefore$
$i^2 = N \therefore$
$i = \sqrt{N}$
$\sqrt{N}$ é inteiro $\rightarrow$ N é quadrado perfeito!

Solução Aceita

```
i=1;
Enquanto i*i <= N:
    Imprime i*i;
    i = i+1;
```

STL

- ```
  #include <algorithm>

  int v[1024], N;

  sort(v,v+N);
  ```
  ordena o vetor v em $O(NlogN)$
- `if (binary_search(v,v+n,t))`
  o elemento t está no vetor (em $O(logN)$).
- http://www.cppreference.com
  ```

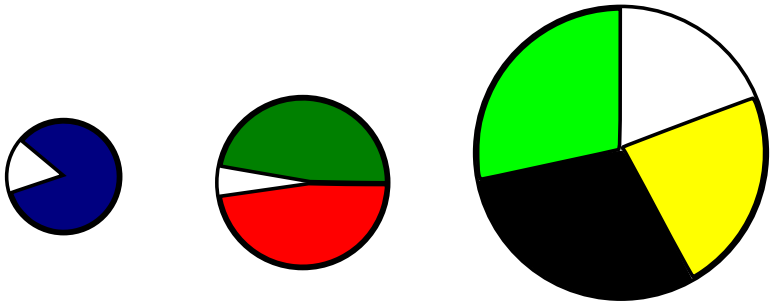
Debug is on the table!

- GDB
- (gcc,g++) -o programa programa.(c,cpp) **-ggdb**
  - **b** *l* - breakpoint na linha *l*
  - **r** - roda o programa
  - **n** - próxima linha
  - **p** *var* - imprime o valor de *var*

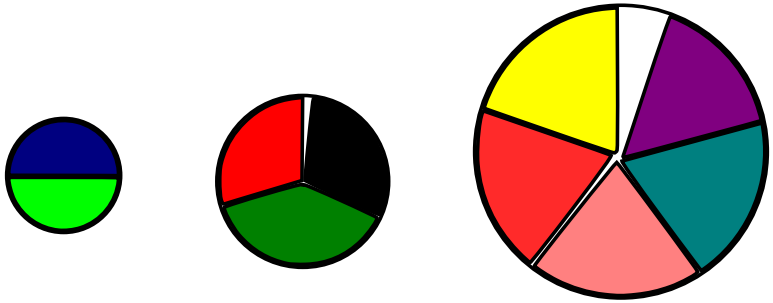
Caderno

- Leve **qualquer** material impresso;
- Códigos prontos de algoritmos frequentes;
- Cópia e digitação dos códigos "no olho" (sem $Ctrl + C + V$!)

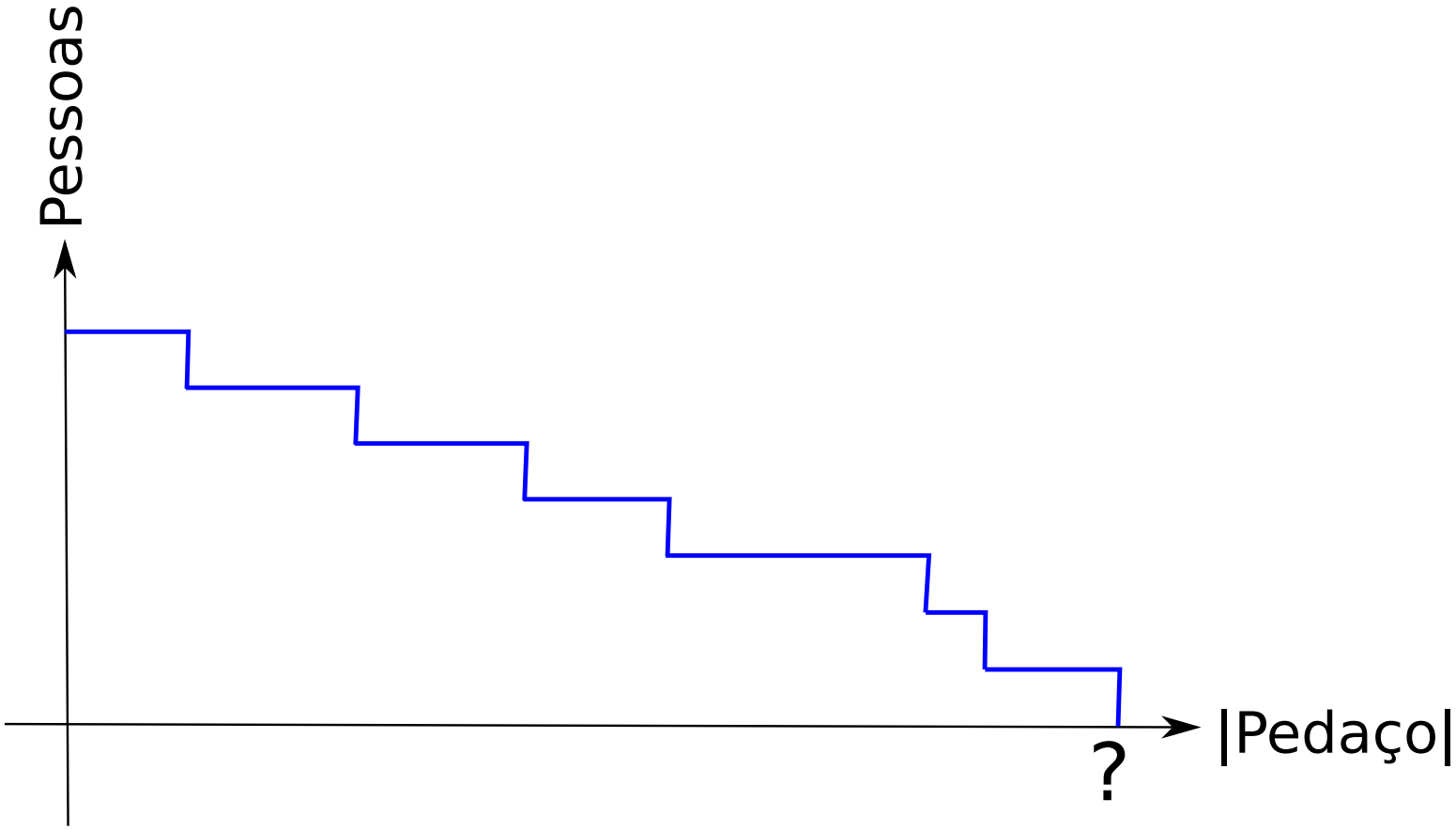
Proporcional

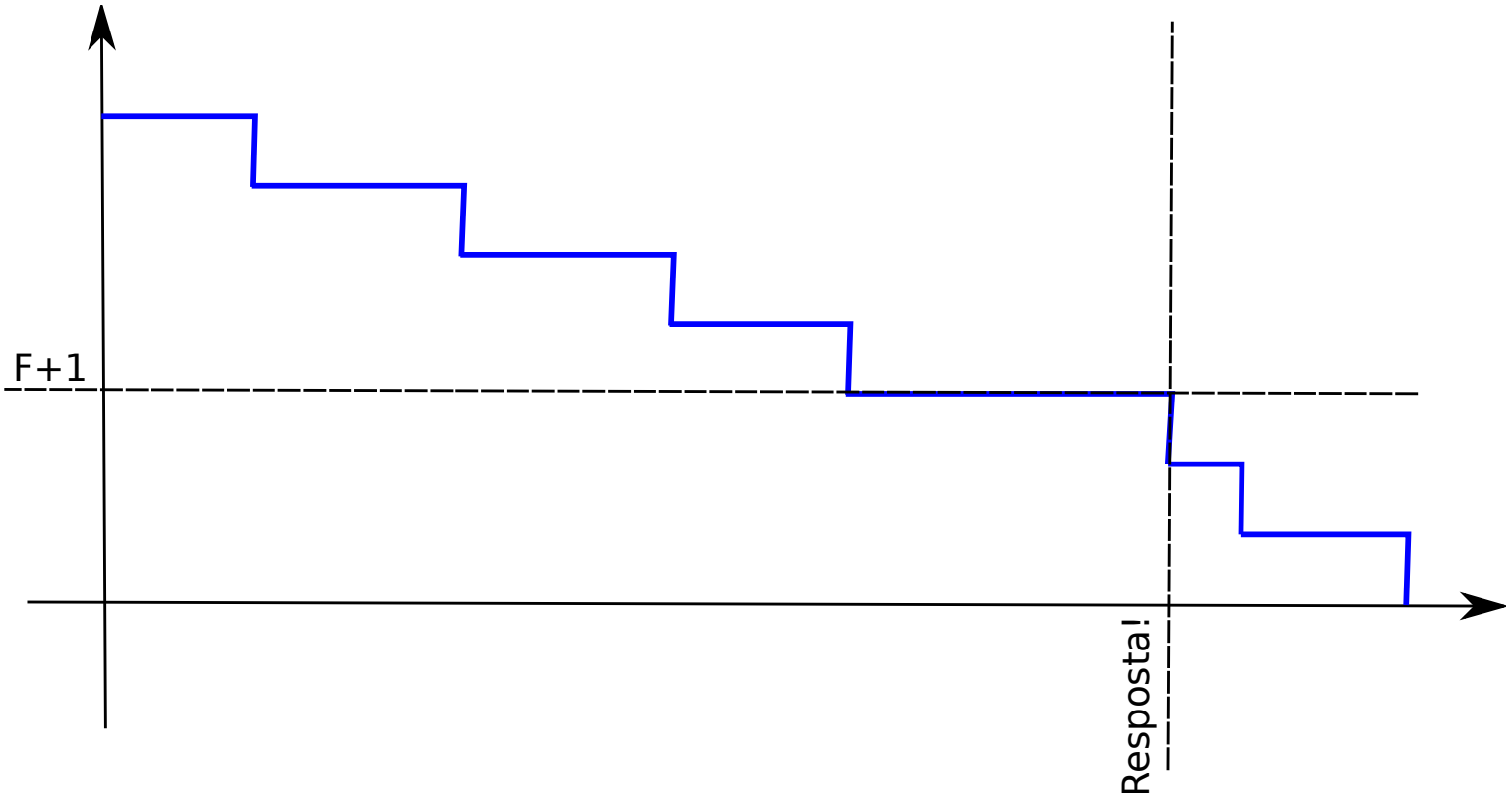


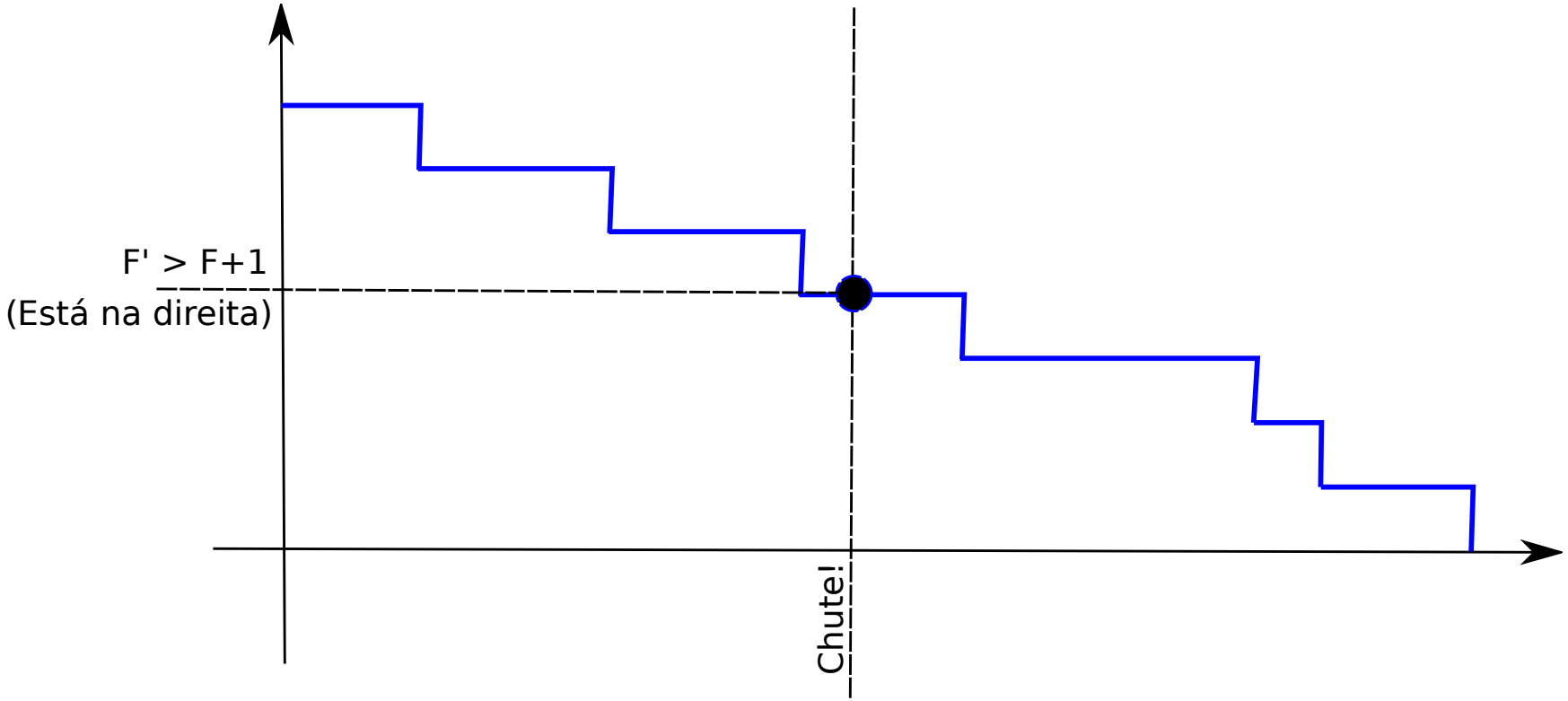
= 6 pessoas

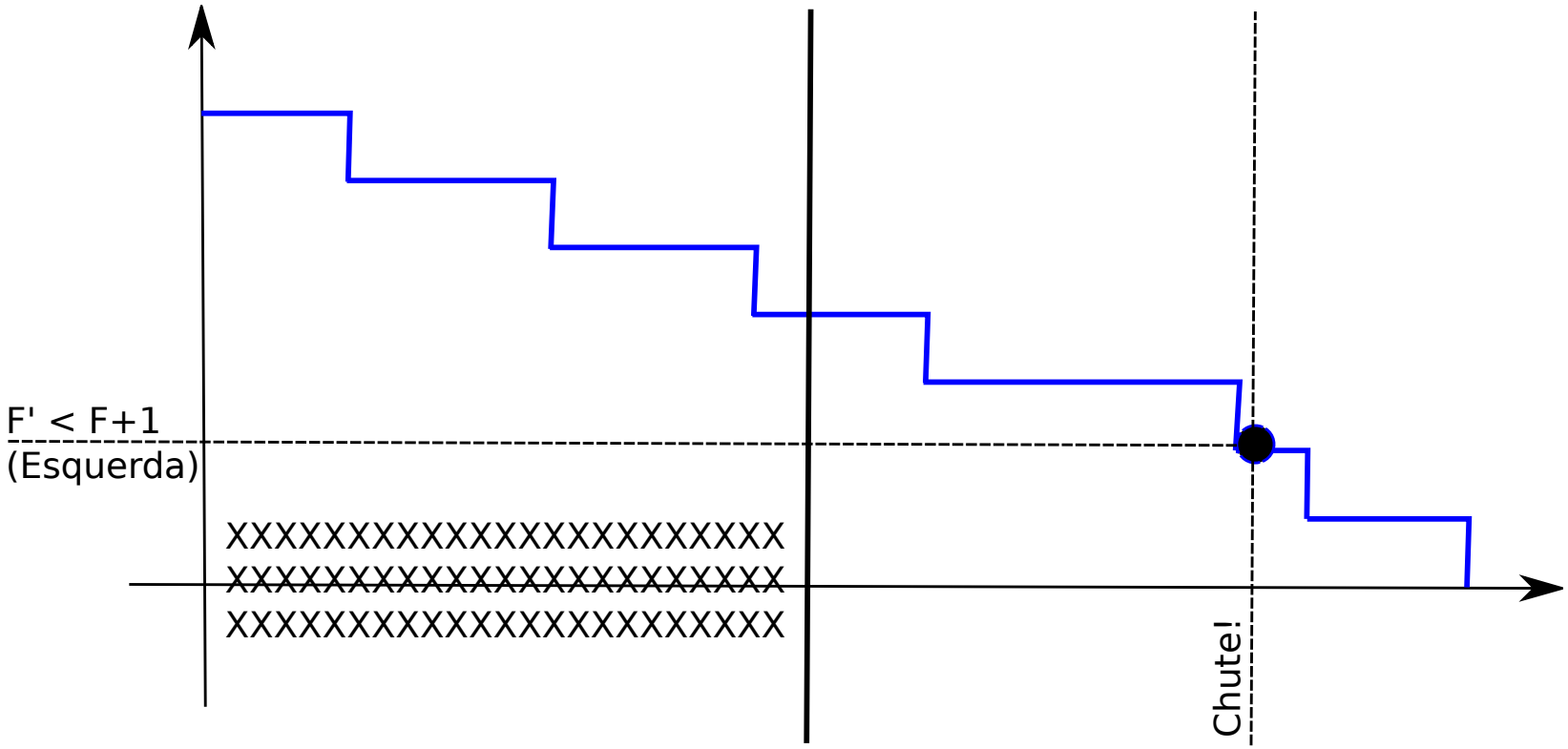


= 10 pessoas

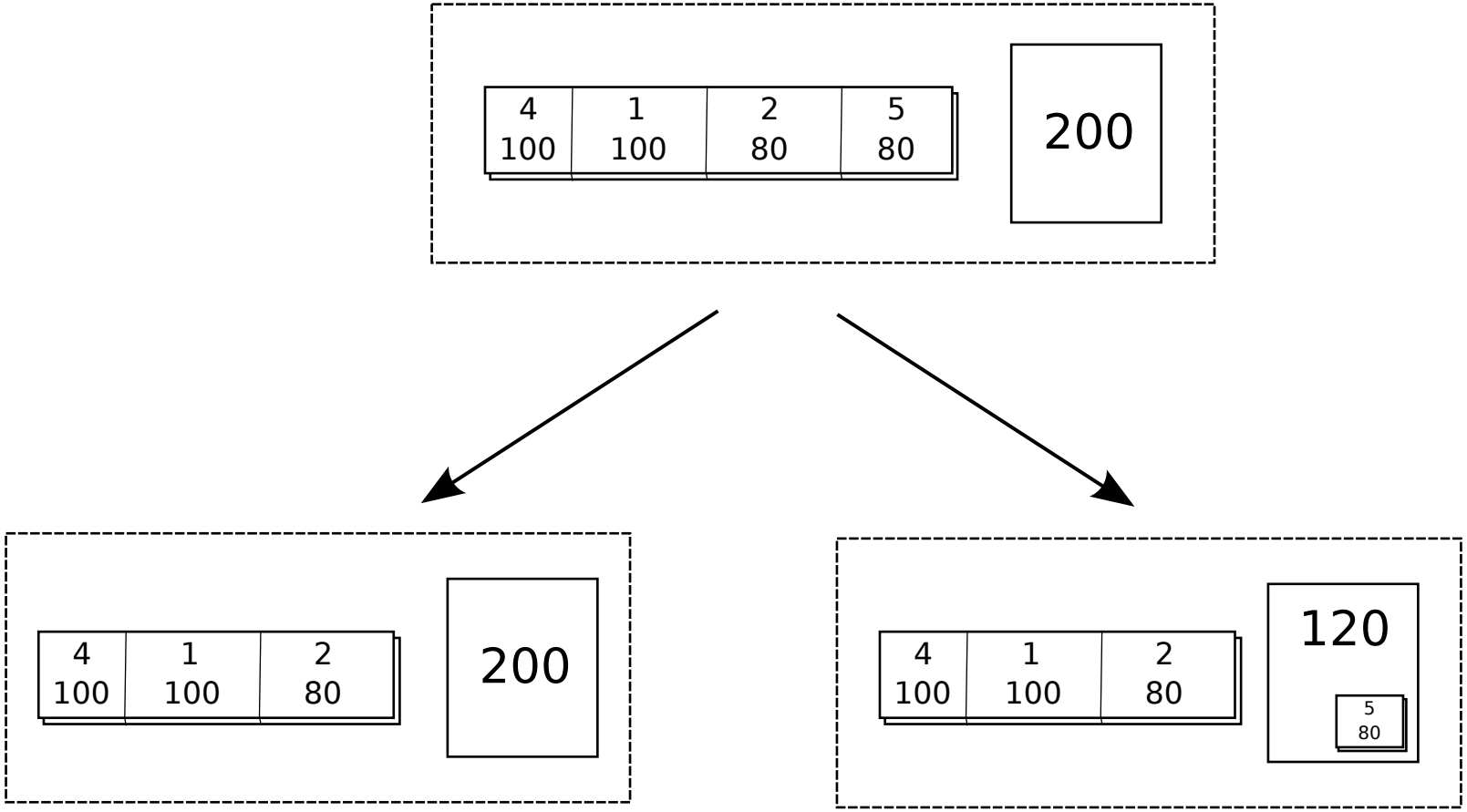

Pessoas
|Pedaço|
?


F+1
Resposta!


F' > F+1
(Está na direita)
Chute!

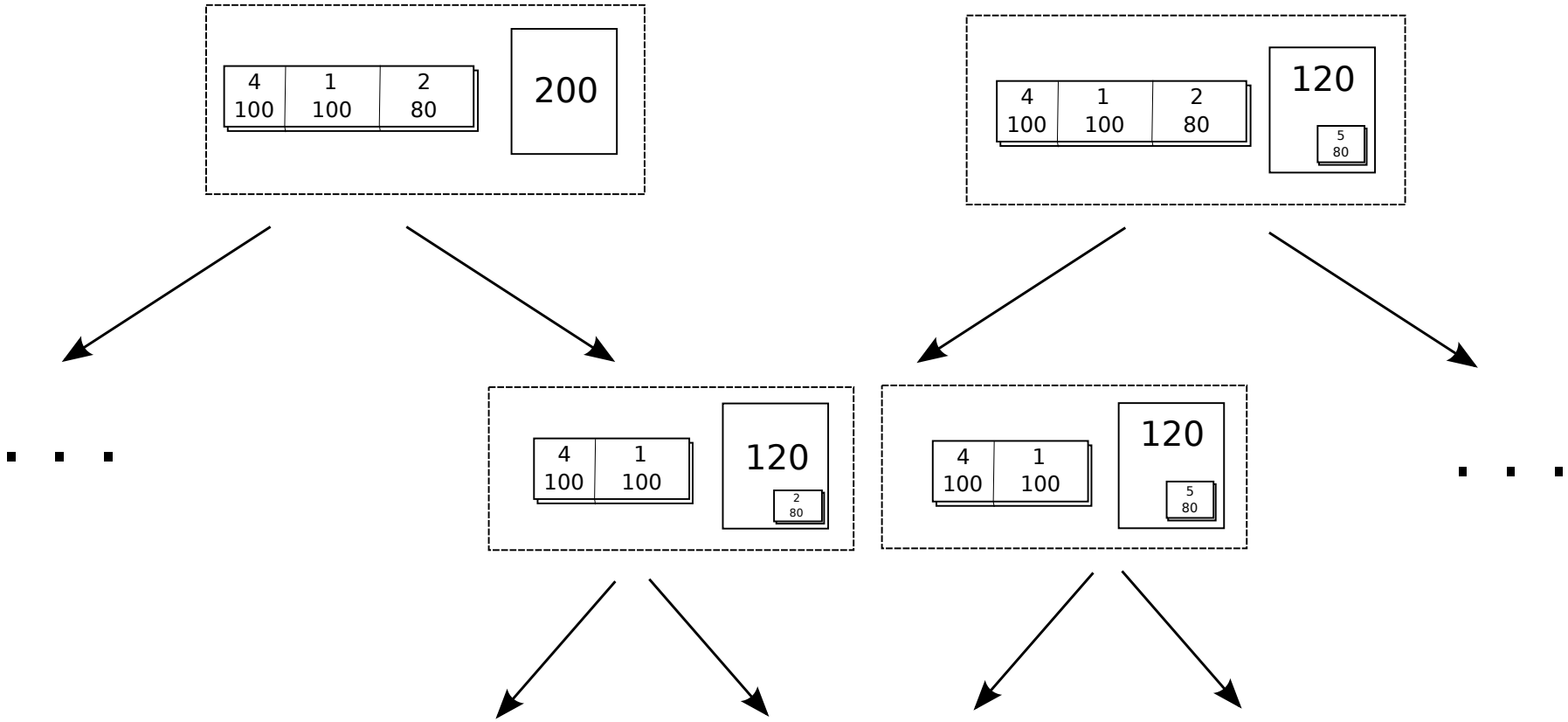

F' < F+1
(Esquerda)
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
Chute!

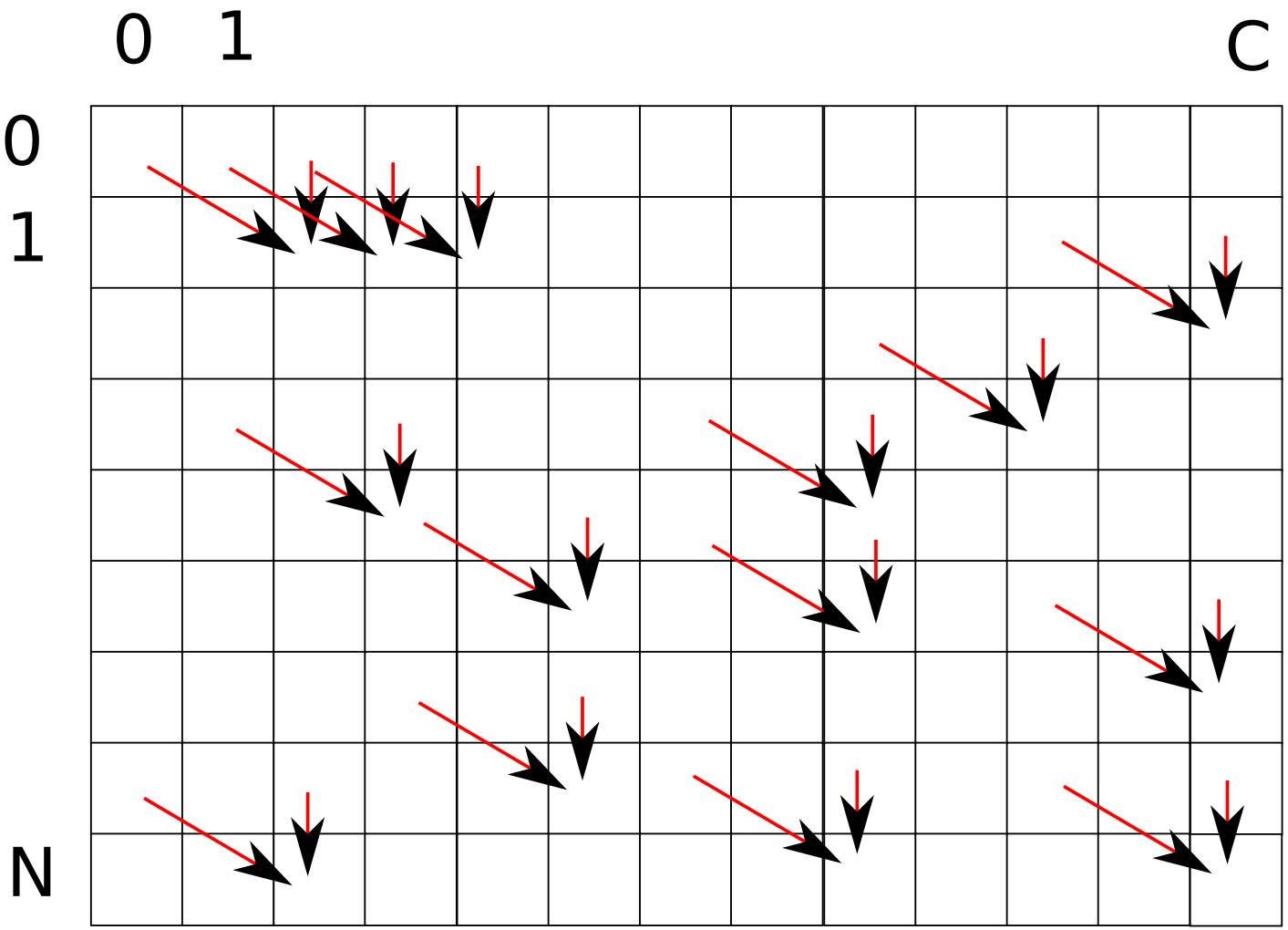
Força bruta



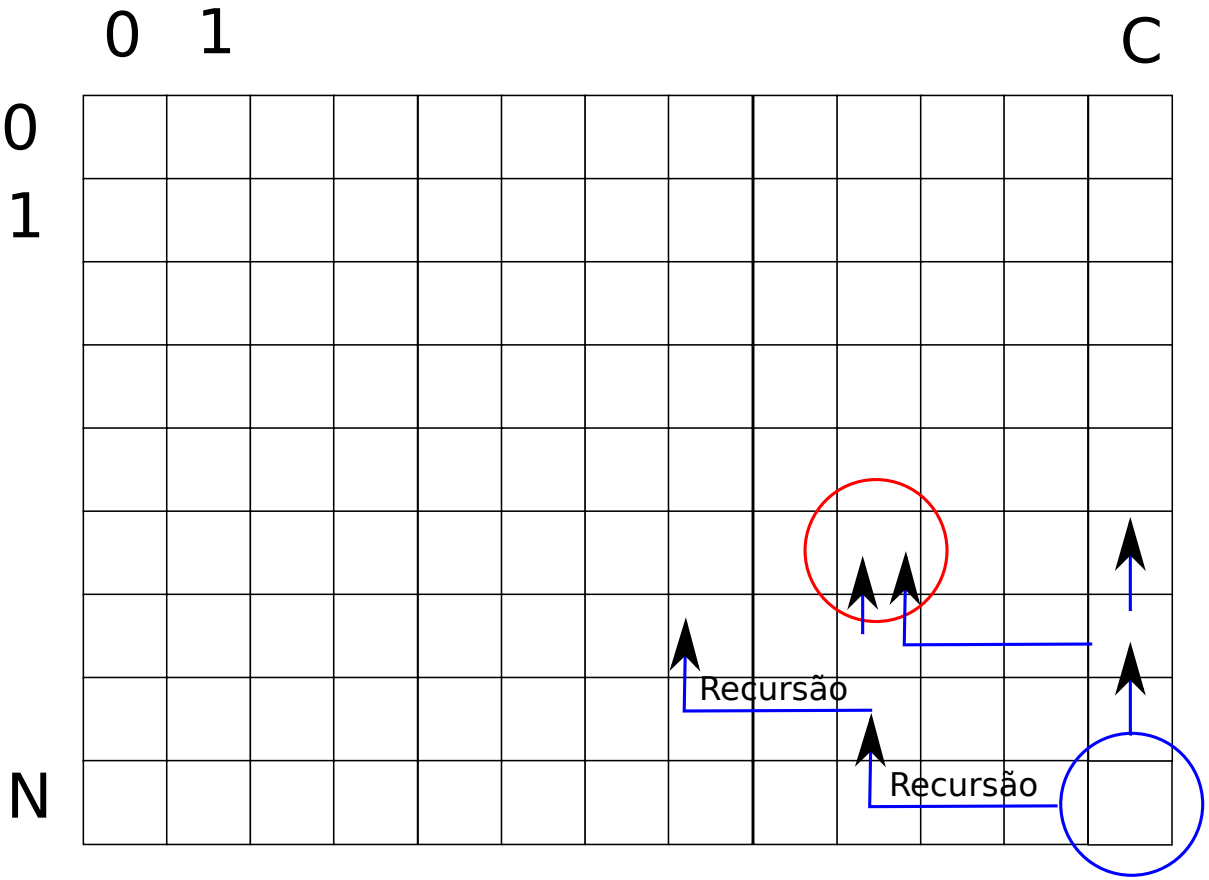
Recorrência

$$resp(N, C) =$$
$$max \begin{cases} resp(N-1, C) \\ valor_N + resp(N-1, C - peso_N) \text{ se } peso_N \leq C \end{cases}$$
$$resp(0, C) = 0$$


4
100
1
100
2
80
200
4
100
1
100
2
80
120
5
80
. . .
4
100
1
100
120
2
80
4
100
1
100
120
5
80
. . .

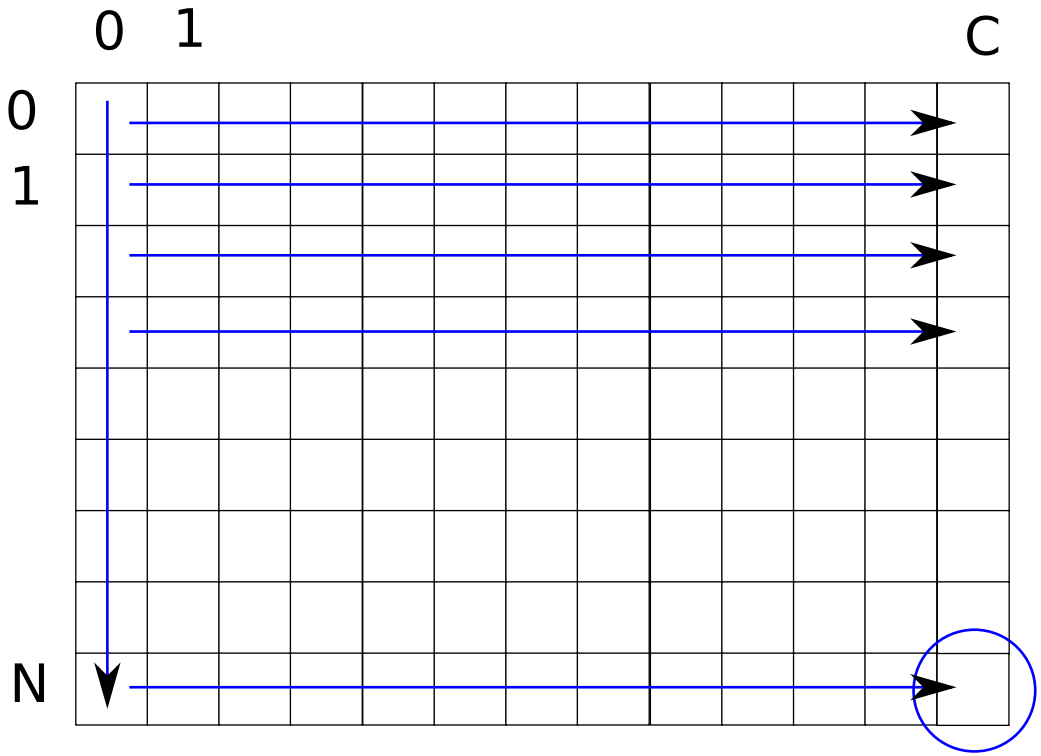

0
1
C
0
1
N

Memorização



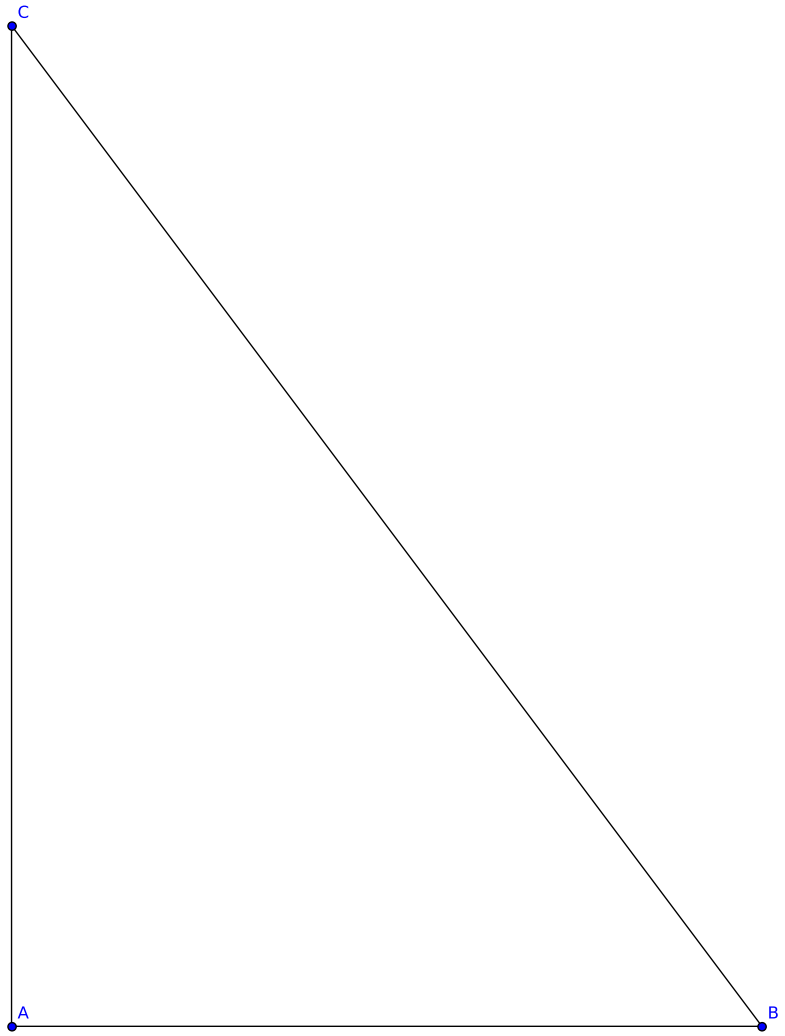
Se já calculado, retorna
Se não, recursão

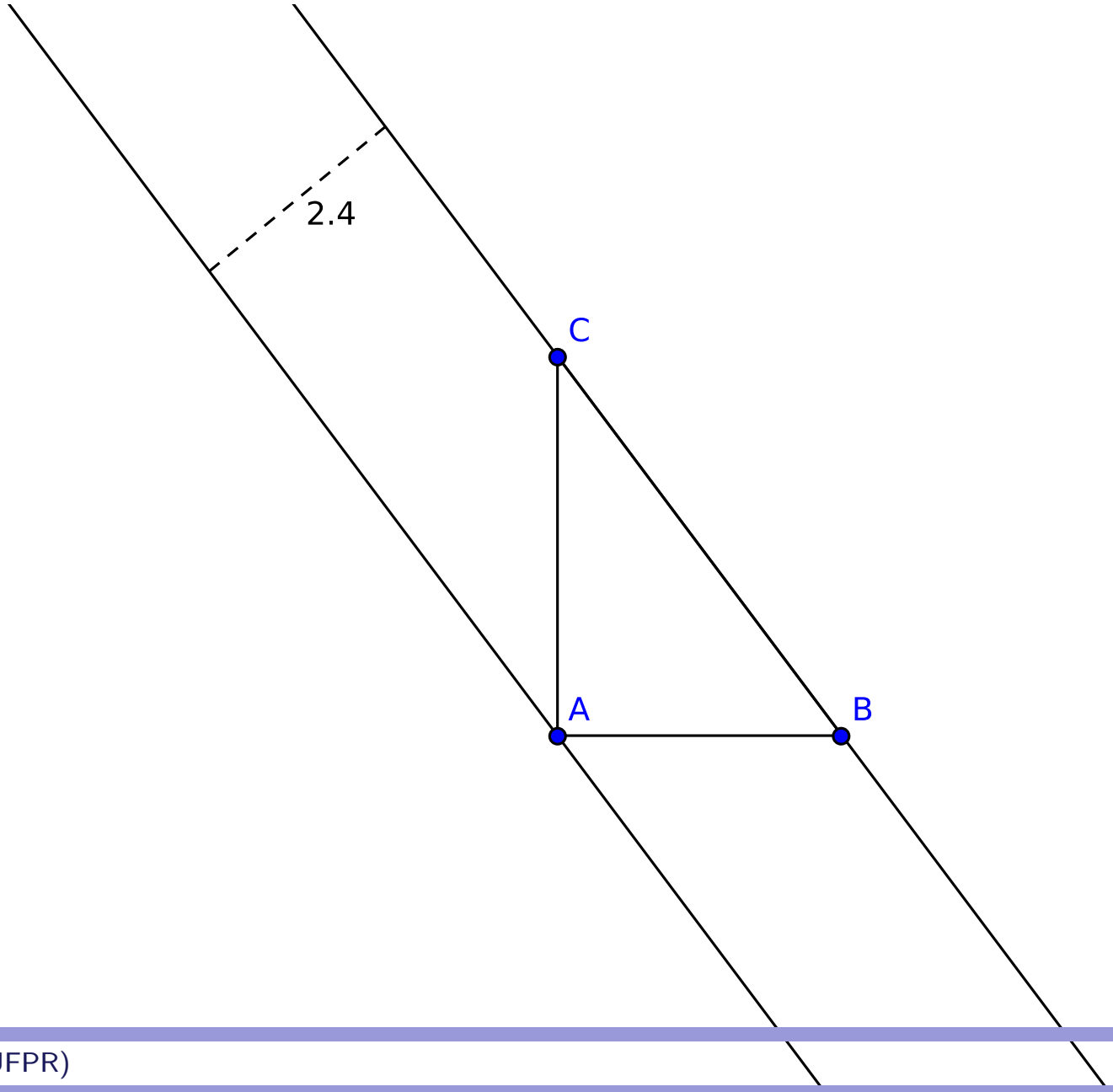
Programação Dinâmica

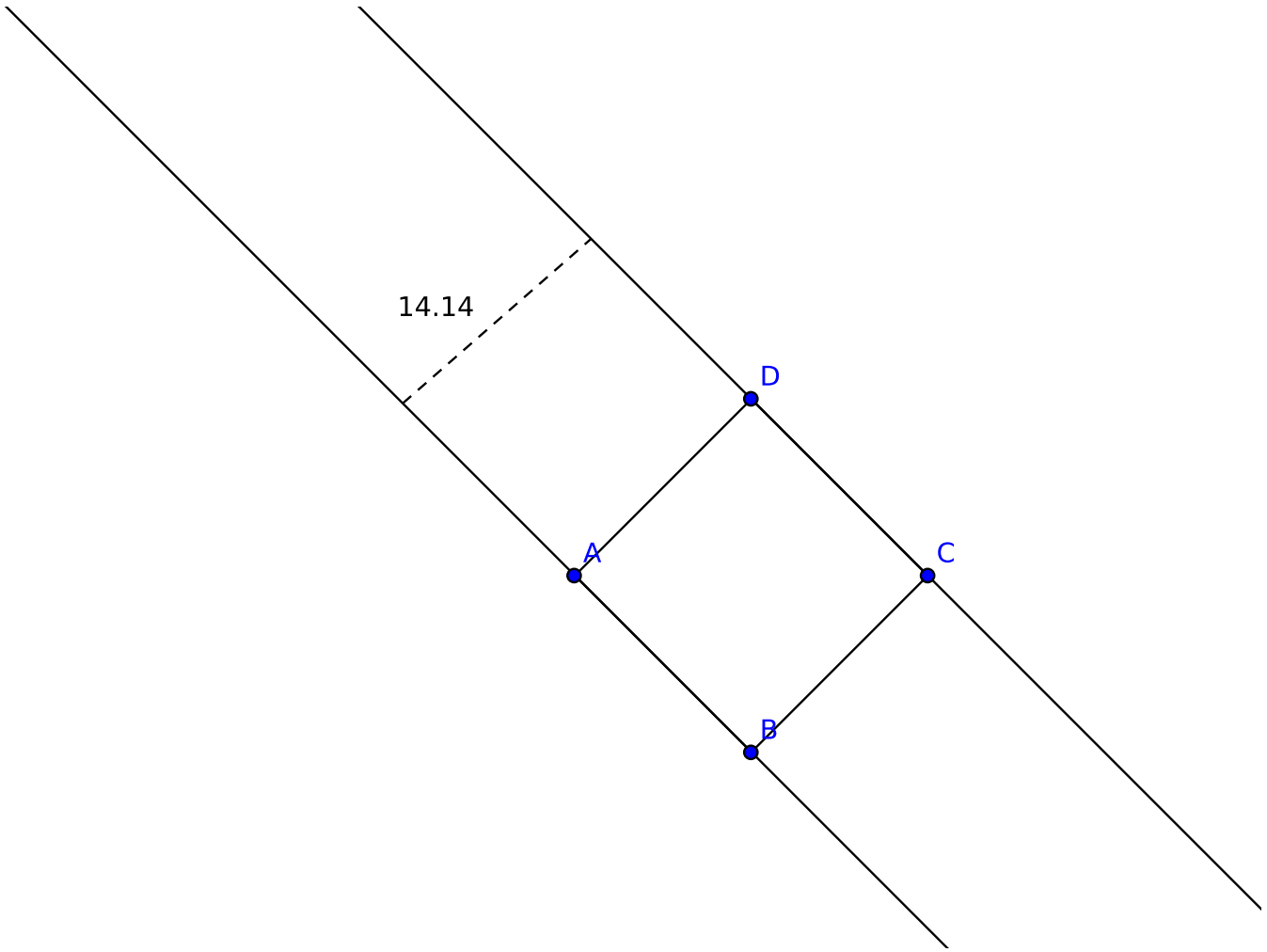


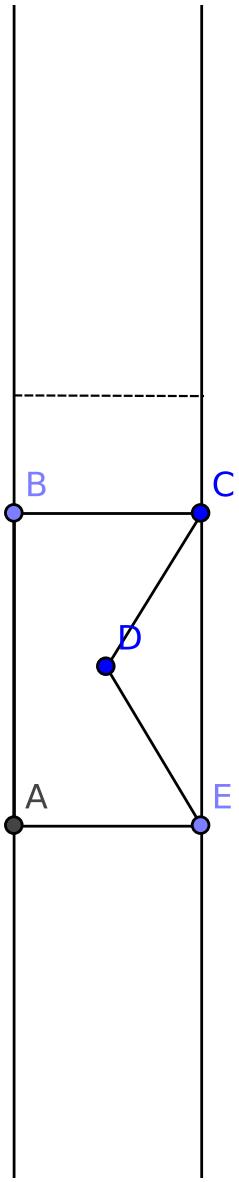
Para n de 0 a N:
Para c de 0 a C:
M[n][c] = ...;

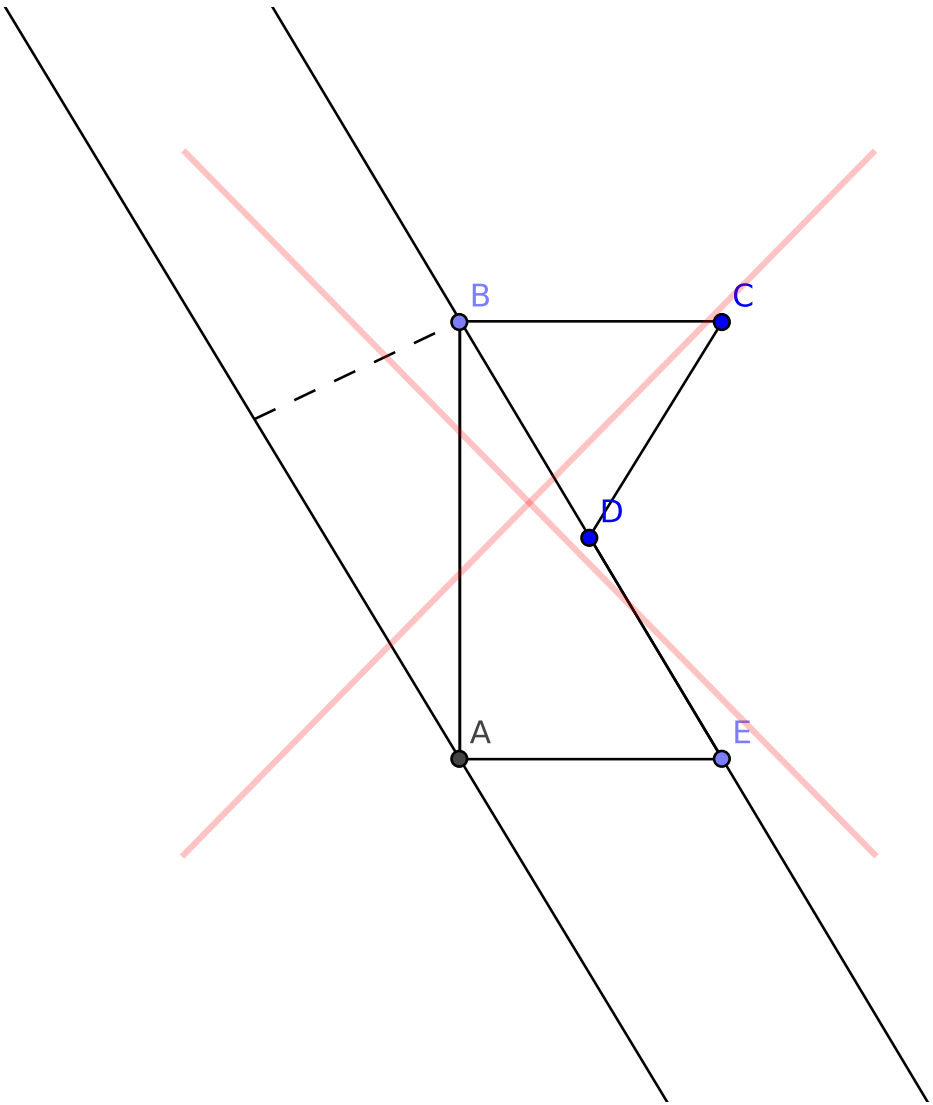
Figura



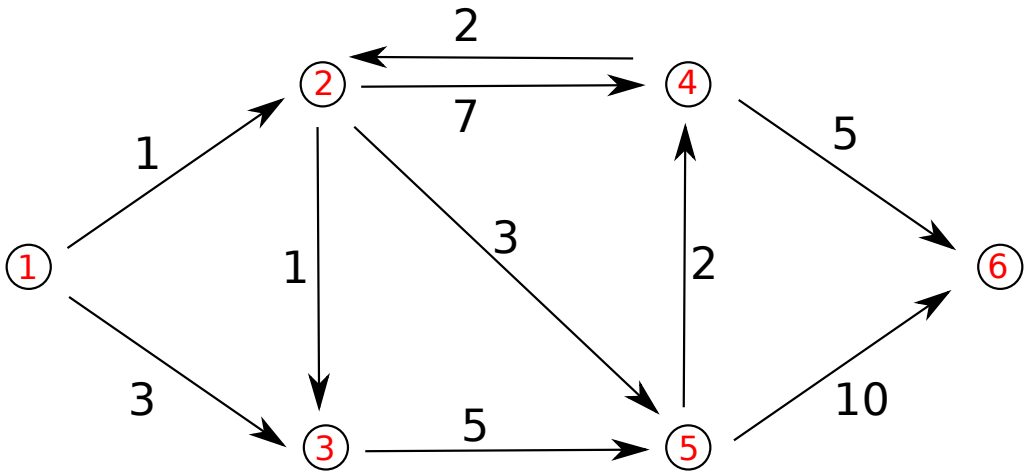

2.4
C
A
B


14.14
D
A
C
B


B
C
D
A
E


B
C
D
A
E

Grafos



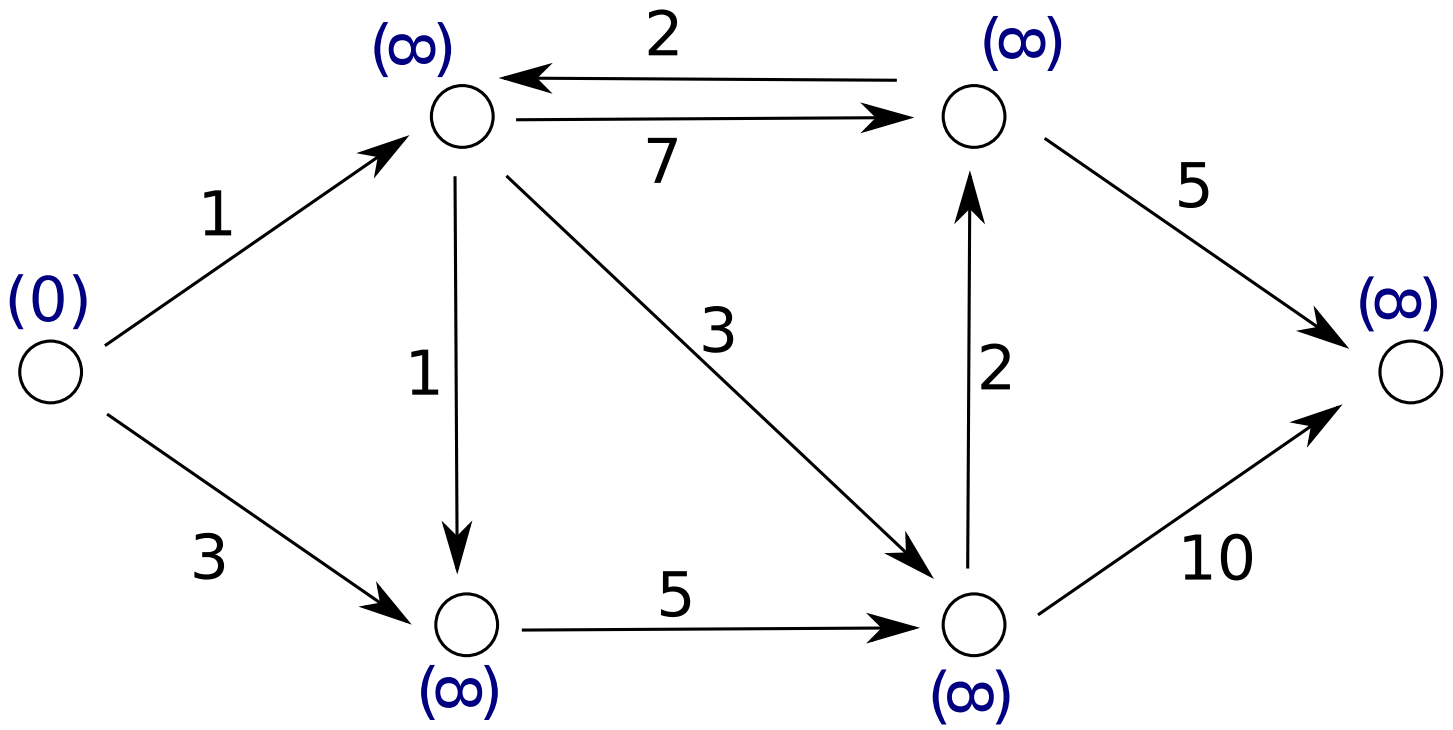
	1	2	3	4	5	6
1	∞	1	3	∞	∞	∞
2	∞	∞	1	7	3	∞
3	∞	∞	∞	∞	5	∞
4	∞	2	∞	∞	∞	5
5	∞	∞	∞	2	∞	10
6	∞	∞	∞	∞	∞	∞

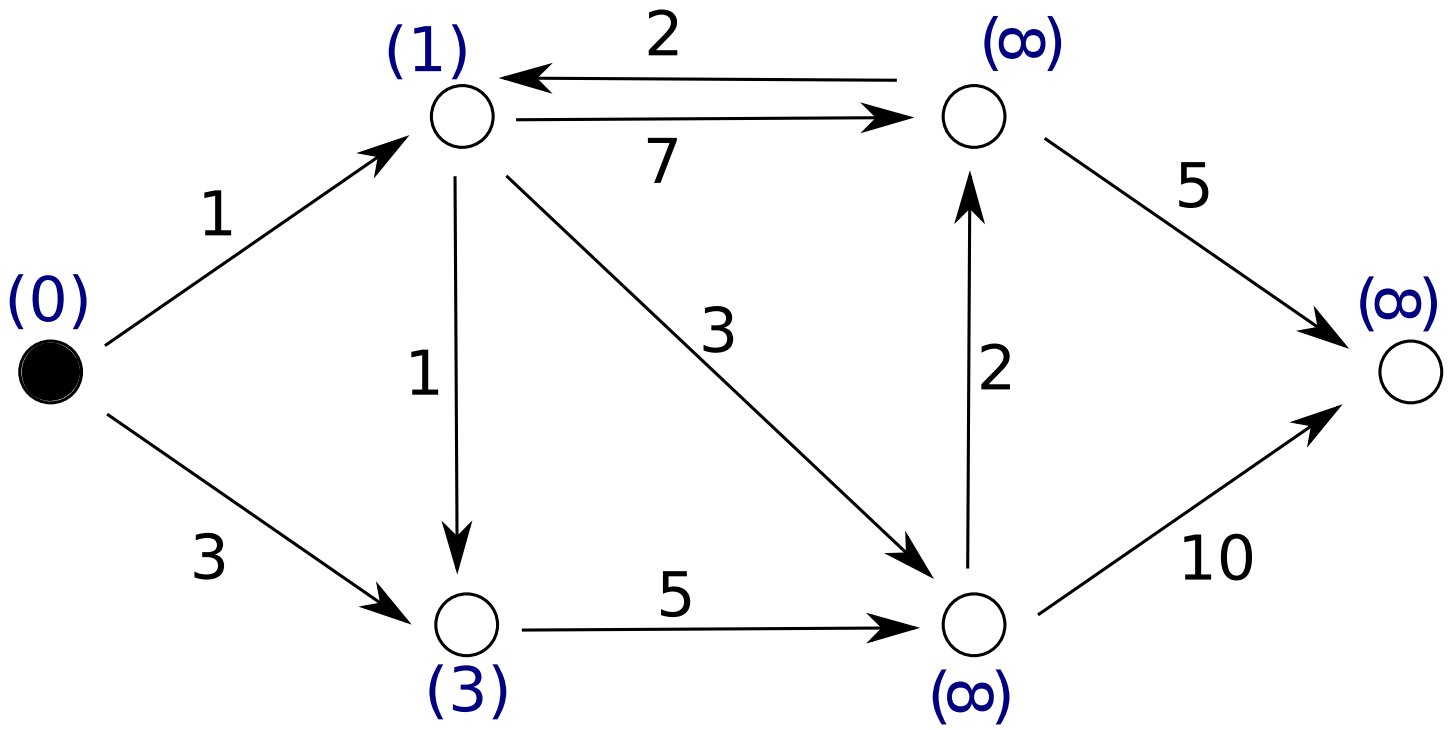
Algoritmo de Dijkstra

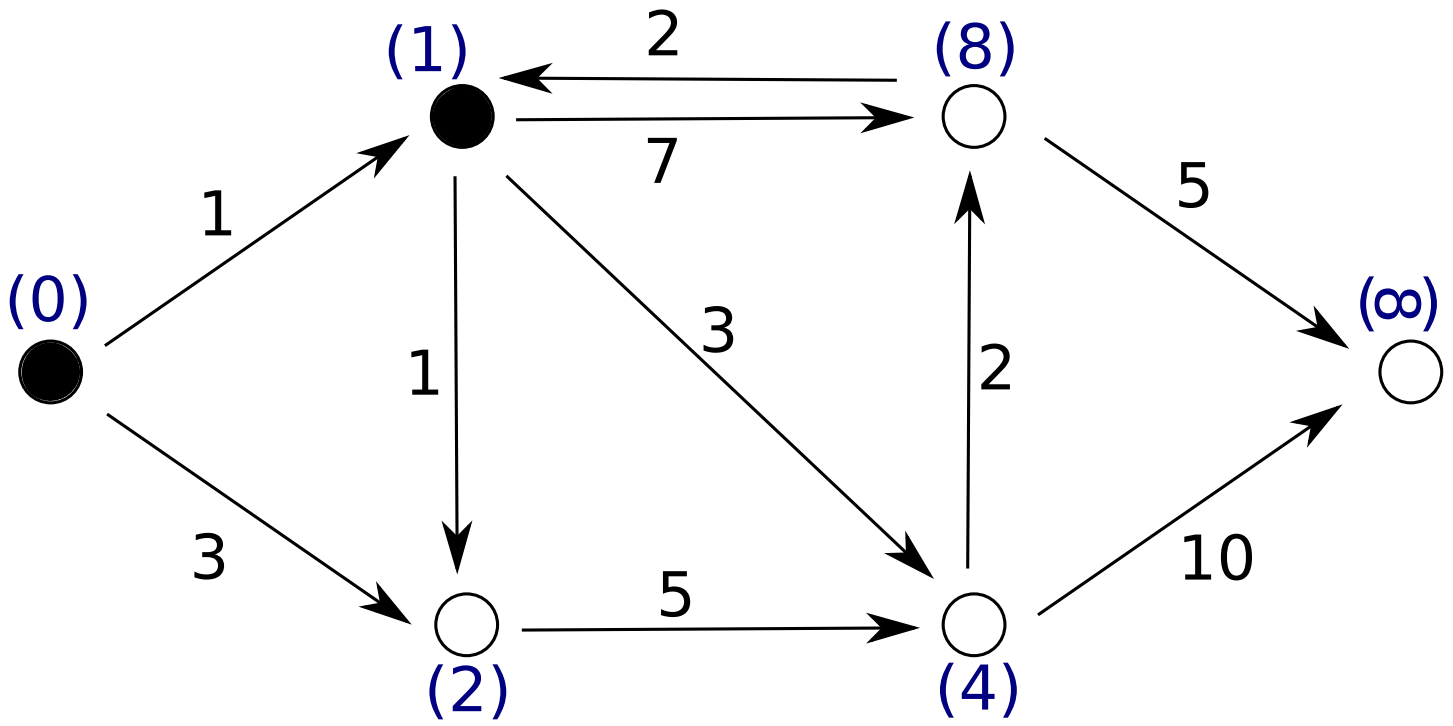
```
Para i de 1 a N:
    dist[i] = INF, fechado[i] = false;
dist[ini] = 0;

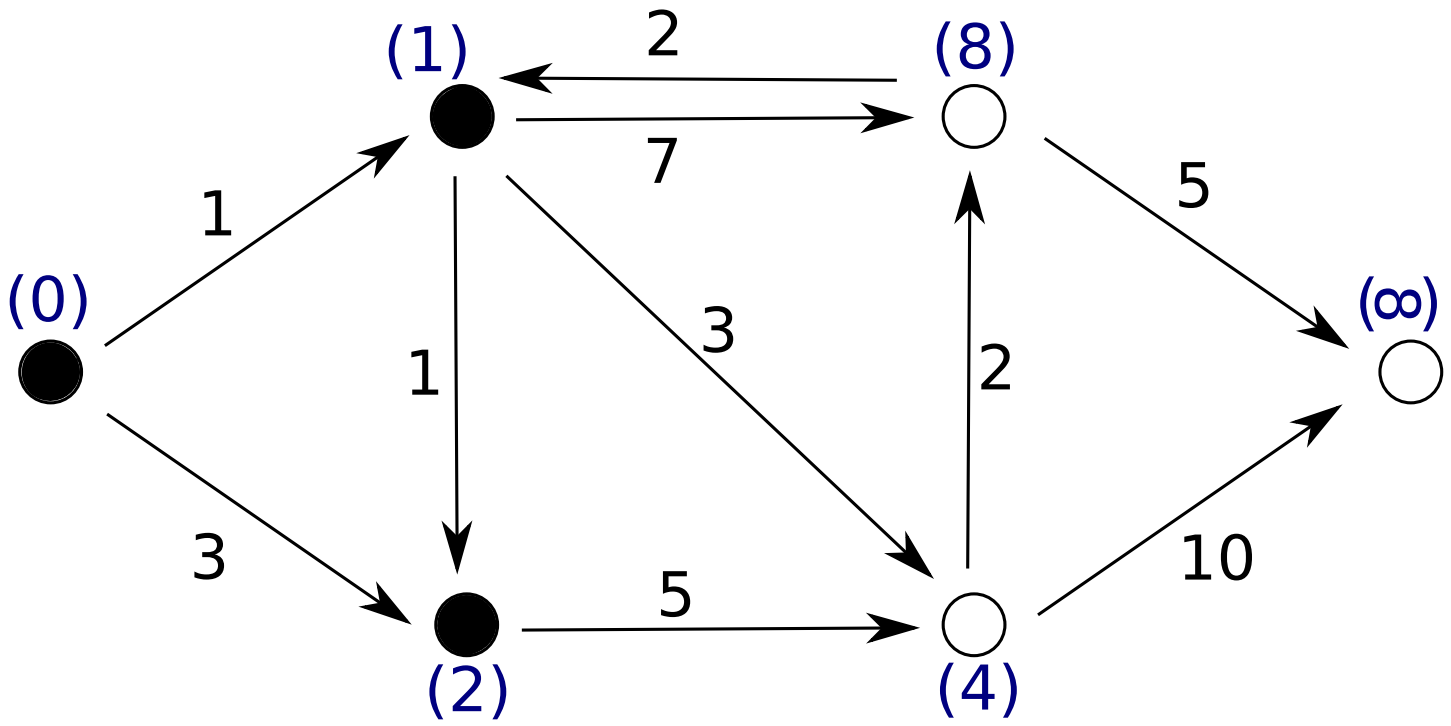
Faça N vezes:
    i = Vertice aberto de menor dist;
    fechado[i] = true;
    Para cada vizinho j de i:
        dist[j] = min(dist[j], dist[i] + M[i][j]);

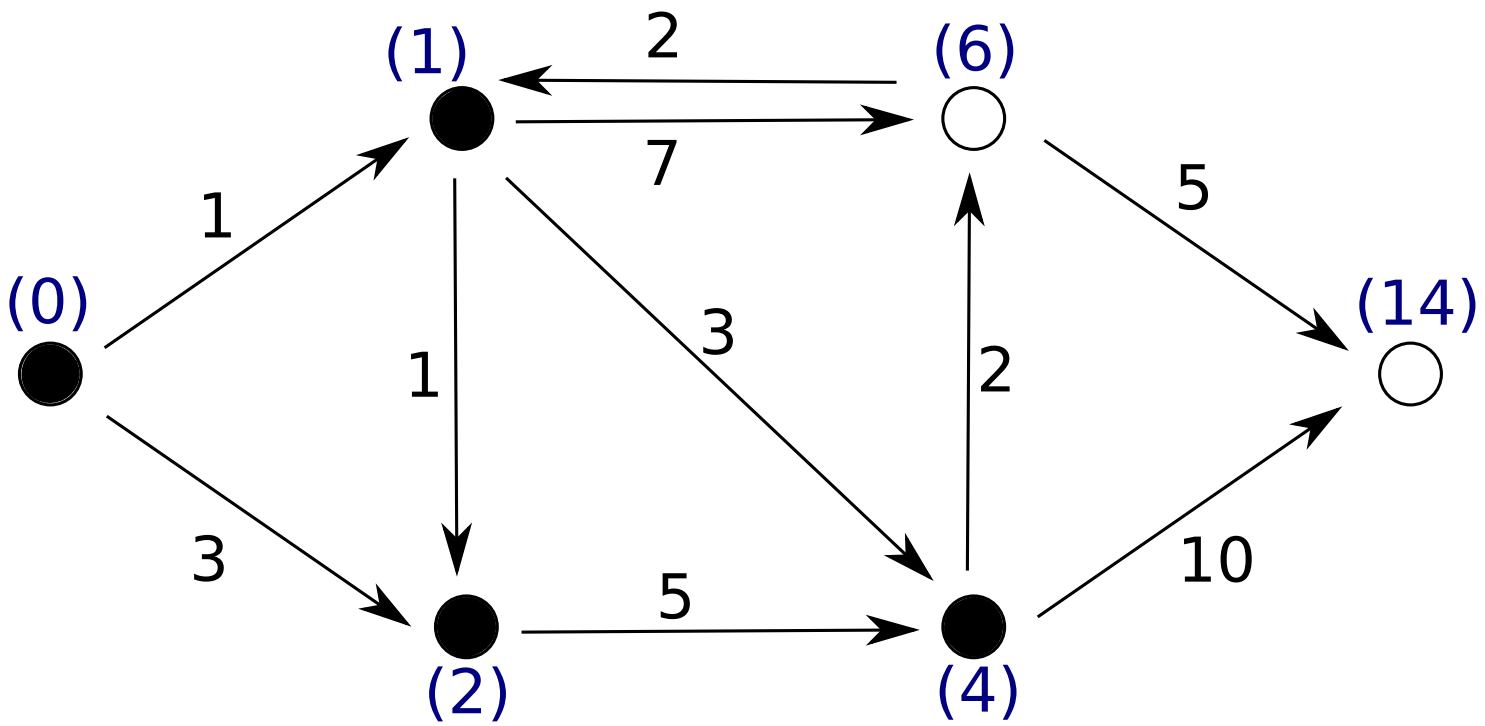
Imprima dist[fim];
```

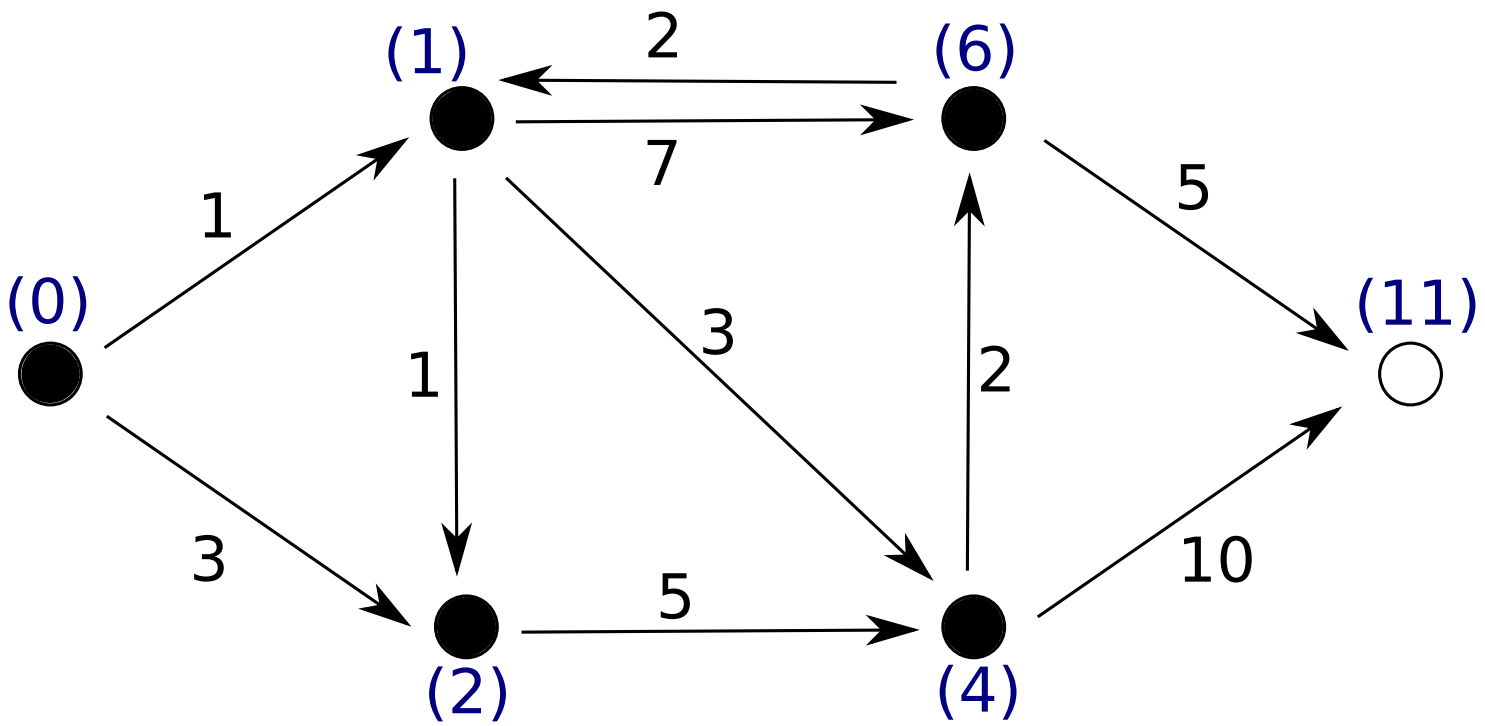

(∞)
2
(∞)
7
1
5
(0)
3
(∞)
1
2
3
10
5
(∞)
(∞)

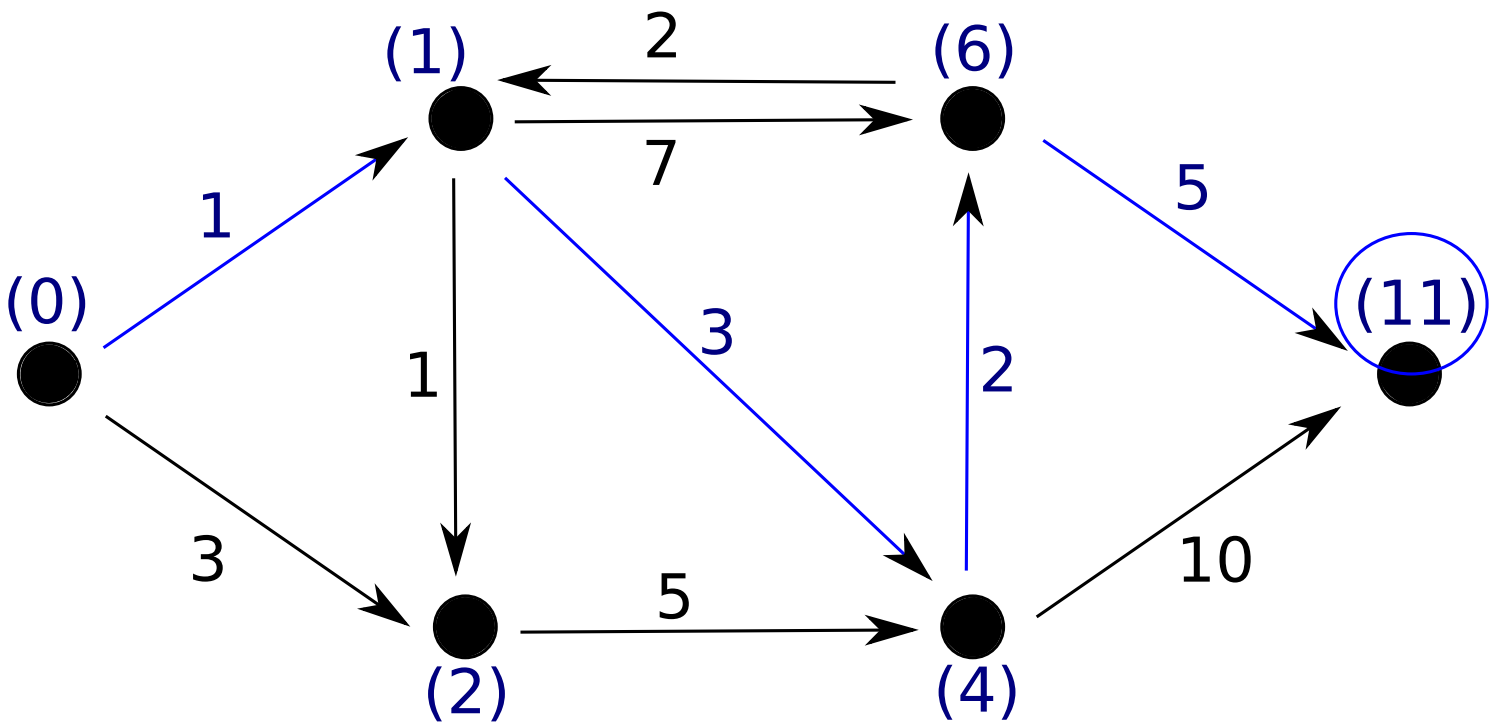

(1)
2
(∞)
7
1
5
(0)
1
3
2
(∞)
3
10
5
(3)
(∞)


(0)
(1)
(2)
(4)
(8)
(∞)
1
3
2
7
1
3
5
2
5
10


(1)
2
(8)
7
1
5
(0)
1
3
2
(∞)
3
5
10
(2)
(4)


(0)
(1)
(2)
(4)
(6)
(14)
1
3
2
7
1
3
5
2
5
10


(0)
(1)
(2)
(4)
(6)
(11)
1
3
2
7
1
3
5
2
5
10


(0)
(1)
(2)
(4)
(6)
(11)
1
3
2
7
1
3
5
2
5
10

Algoritmo Simples

```
Igual(string S, string R):
    Se |S| != |R| então retorna false;
    Para i de 1 a |S| Faça
        Se S[i] != R[i] então
            retorna false;
    retorna true;

Para i de 0 a |palheiro|-1 faça
    Se Igual(agulha, palheiro:i) enão
        Imprime i
```

Hashing

- "Xunxo"
- String de tamanho linear $\rightarrow$ número de tamanho constante
- "*cano*" $= 90783$
- *#define hash unsigned long long int*

Hashing

- $132 = 1 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$
- $ana = a \times B^2 + n \times B^1 + a \times B^0$
- $a = 1, b = 2, ...$
- Obrigatório: $B \geq 26$
- Recomendado: B ímpar/primo, $B = 31$

- $cano = c \times B^3 + a \times B^2 + n \times B^1 + o \times B^0$
- $= 3 \times 31^3 + 1 \times 31^2 + 14 \times 31 + 15 = 90783$

Vetor de prefixos

E	X	C	A	N	O	A
5	179	5552	172113	5335517	16540 1042	512743 2303

$ex = e \times B + x$

$exc = ex \times B + c$

...

Extraindo *substrings*

E	X	C	A	N	O	A
5	179	5552	172113	5335517	16540 1042	512743 2303

EXCANO -
EX0000 =
00CANO

$$165401042 - 179 \times B^4 =$$
$$165401042 - 165310259 = 90783$$

Outros temas frequentes

- Árvore de Segmentos/BIT/Treaps
- Kruskal/Bellman/Fluxo/Matching/etc
- Vetor de Sufixos/KMP/Aho-Corasick/etc
- Fecho Convexo/Enclosing Circle/etc
- Busca Ternária
- Diversos outros