Algoritmi APPELLO 15/07

Esercizi 1-2

1) Dato un array $A[1 \dots n]$,
i è un GAP se $A[i+1] - A[i] \geq 2$

- Mostrare per induzione che

$\forall A[1 \dots n]$ $(n \geq 2)$,
SE $A[n] - A[1] > n$ ALLORA ESISTE ALMENO UN GAP.

Dim:

CASO BASE: $(n=2)$

$$A[2] - A[1] > 2 \Rightarrow 1 \text{ è un GAP.}$$

CASO INDUTTIVO: $(n > 2)$

Ipotesi induttiva: $A[n] - A[1] > n$

Occorre dimostrare che:

se $A[n+1] - A[1] > n+1$ allora $A[1 \dots n+1]$ ha almeno un gap.

$$A[n] - A[1] = A[n+1] - A[1] - (A[n+1] - A[n])$$

2 possibilità:

- $A[n+1] - A[n] \geq 2$

  e si conclude che n è un gap.

OPPURE

- $A[n+1] - A[n] < 2$.

In questo caso:

$$A[n] - A[1] =$$

$$= \underbrace{A[n+1] - A[1]}_{\substack{> n+1 \\ \text{per ipotesi}}} - \underbrace{\left(A[n+1] - A[n] \right)}_{\leq 1}$$

$$\Rightarrow A[n] - A[1] > n$$

e quindi, per ip. induttiva

$A[1 \ldots n]$ contiene un gap

$\Rightarrow A[1 \ldots n+1]$ contiene un gap.

```
GAP(A, p, q) // restituisce un gap in A[1...n]
             // A è tale che
             //    A[n] - A[1] > n

    if (q - p = 1)
        return p
    else
        r = ⌊(p + q)/2⌋
        if (A[q] - A[r] > q - r + 1)
            return GAP(A, r, q)
        else
            return GAP(A, p, r)
```

COMPLESSITÀ:

$$T(n) = T\left(\frac{n}{2}\right) + c$$

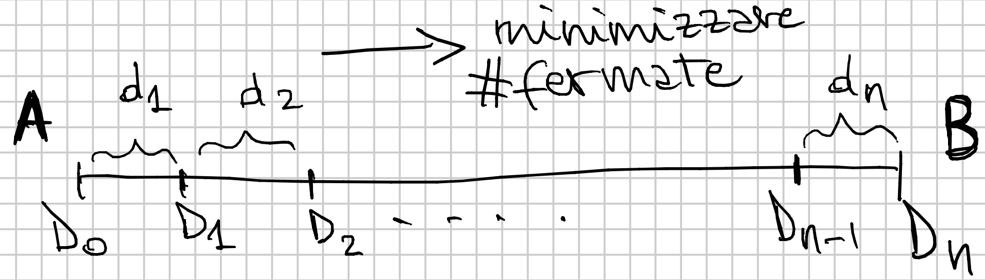
$a = 1$
$b = 2$ $\quad n^{\log_2 1} = n^0 = 1$

$f(n) = c = \Theta(1)$

CASO 2 M.T.

$$\Rightarrow T(n) = \Theta(\log n)$$

2



d: autonomia

Si cerca una soluzione ottima

(un sottoinsieme di $\{D_0, D_1, \dots, D_n\}$ tale che la distanza tra due fermate successive sia $\leq d$)

$$d_{i,j} := \sum_{h=i+1}^{j} d_h \quad (i<j)$$

soluzione del tipo: $\{D_{x_0}, D_{x_1}, D_{x_2}, \dots D_{x_k}\}$

$$\forall j = 0 \dots k-1 \qquad d_{x_j, x_{j+1}} \leq d$$

costo soluzione:

$$c(S) = k-1 \quad (\#\text{fermate})$$

1. <u>SOTTOSTRUTTURA OTTIMA</u>

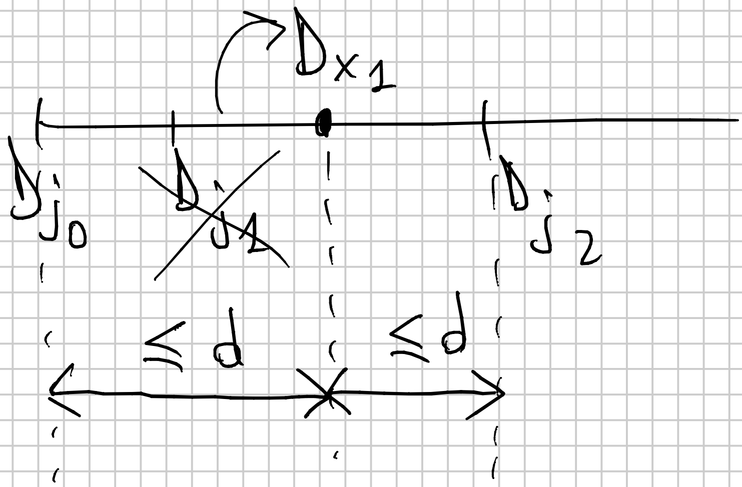
Bisogna mostrare che ogni soluzione ottima a questo problema "contiene" al suo interno soluzioni ottime di sottoproblemi.

Sia il problema $D_0, D_1, \dots D_n$
e si supponga $\{D_{x_0}, D_{x_1}, \dots, D_{x_k}\}$
essere una soluzione ottima.
Allora il sottoproblema
$$D_{x_1}, D_{x_1+1}, \dots, D_n$$
ha come soluzione ottima
$$S_1 = \{D_{x_1}, D_{x_2}, \dots, D_{x_n}\}.$$
Si dimostra per assurdo. Se esistesse S_2 migliore di S_1, allora $D_{x_0} \cup S_2$ sarebbe migliore di S per il problema. ASSURDO

SCELTA GREEDY

problema $D_0, D_1, \dots, D_n$
$$D_{x_0} = D_0$$
$$x_1 = \max\{J \mid d_{0,J} \leq d\} \leftarrow \text{SCELTA GREEDY}$$
Bisogna dimostrare che la scelta greedy porta sempre a una sol. ottima
Sia $D_{j_0}, D_{j_1}, \dots, D_{j_K}$ una sol. ottima, allora posso sostituire D_{j_1} con D_{x_1}: $D_{j_0}, D_{x_1}, \dots, D_{j_K}$.

$x_1 \geq j_1$ (per la scelta greedy)



Stop(d, n)   d[1 ... n]

```
dist = d[1]
for i = 2 to n
    if(dist + d[i] ≤ d)
        dist = dist + d[i]
        S[i-1] = 0
    else
        S[i-1] = 1
        dist = d[i]
return S
```

COMPLESSITÀ: $\Theta(n)$

SOLUZIONI TRATTE DALLA CORREZIONE IN AULA DEL 24/07.