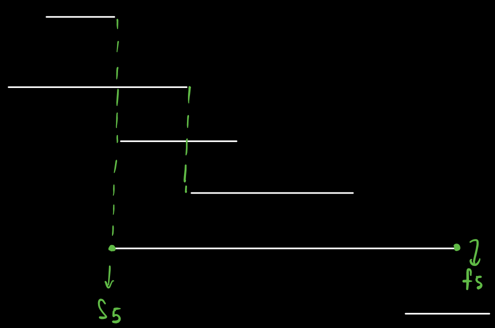
2^k combinazioni {algoritmo f. bruta: $\Omega(2^n)$
↑
OTTIMIZZAZIONE COMB.
↑

ESERCIZI SULLA PROGRAMMAZ. DINAMICA.

PROGRAMMAZIONE INTERVALLI PESATI:

Trovare un insieme di attività mutualmente compatibili, e che massimizzi il "valore" che apportano.

- ORDINATI RISPETTO AL TEMPO DI FINE.
- ALCUNI SONO "COMPATIBILI", ALTRI NO.



S in questo caso: $\{1, 3, 6\}$

i	$p(i)$	v_i
1	0	10
2	0	2
3	1	8
4	2	1
5	1	1
6	4	3

↓ PRIMO COMPATIBILE

ISTANZA:

$X = \{1, \dots, n\}$ insieme di n attività $n \in \mathbb{N}$

$\forall$ attività $i \in X$ sono associati:

* S_i, tempo di inizio;
* F_i, tempo di fine;] $S_i \leq f_i$ $[S_i, f_i)$
* V_i, valore;

COMPATIBILITÀ INTERVALLI: $[S_i, f_i) \cap [S_j, f_j) = \emptyset$
↓
"non si sovrappongono"

$A \subseteq \{1, \dots, n\}$ contiene attività mut. comp. se $\forall i, j \in A$, $i \neq j$, i e j compatibili.

Compatibilità intervalli $\simeq$ vincoli sulla regione ammissibile.

FUNZIONE COMP: Funzione $\mathcal{P}(\{1 \dots n\}) \to \{true, false\}$

$\forall (A) \subseteq \{1 \dots n\}$ $\quad comp(A) = \begin{cases} TRUE & \text{se } [S_i, f_i) \cap [S_j, f_j) = \emptyset \\ FALSE & \text{altrimenti} \end{cases}$

→ UTILITÀ: "sintetizzare", "migliorare" il problema.

FUNZIONE VALUE: Funzione $\mathcal{P}(\{1 \dots n\}) \to \mathbb{R}$

$\forall A \in \mathcal{P}(\{1 \dots n\})$ $\quad \forall A \subseteq \{1 \dots n\}$ $\quad V(\{A\}) = \begin{cases} \sum_{i \in A} V_i & \text{SE } A \neq 0 \\ 0 & \text{SE } A = 0 \end{cases}$

ESEMPIO:
1) $A = \{1, 3, 6\}$ $\quad Comp(A) = TRUE$ $\quad V(\{A\}) = 21$
2) $A = \{2, 4, 6\}$ $\quad Comp(A) = TRUE$ $\quad V(\{A\}) = 6$
3) $A = \{1, 2, 3, 4, 5, 6\}$ $\quad Comp(A) = False$

FORMULAZIONE IN TERMINI NON RICORSIVI

ISTANZA: $n \leftrightarrow X = \{1, \dots n\}$ $\quad \forall i \in \{1 \dots n\}$ $\quad S_i, F_i, V_i$

Soluz : $S \subseteq \{1 \dots n\}$ t.c:

1) ~~$Comp(S) = true$~~ → Inclusa nella 2ª

2) $V(S) = \max \{V(A)\}$; $A \subseteq \{1 \dots n\}$ t.c. $V(A) = true$;

FORMULA RICORSIVA:

* SOTTOPROBLEMI ?

- problema completo: $X = X_n = \{1 \dots n\}$ $S_n \subseteq X_n$ t.c. $V(S_n) = \max_{\substack{A \subseteq X_n \\ Comp(A) = true}} \{F(A)\}$

- problema "più facile" = $X_{n-1} \dots X_{n-2} \dots X_0 = X_{n-k}$, finché $n-k \geq 0$ sono tutti i sottoproblemi.

$= S_{n-1} \subseteq X_{n-1}$ t.c. $V(S_{n-1}) = \max_{\substack{A \subseteq X_{n-1} \\ comp(A) = true}} \{V(A)\}$.

IL GENERICO SOTTOPROBLEMA IDENTIFICATO DA:

$i \in \{0, 1 \dots, n\}$

ISTANZA $X_i = \{1, 2 \dots i\}$

SOLUZ. $S_i \subseteq X_i$ t.c. $V(S_i) = \max_{\substack{A \subseteq X_i \\ comp(A) = true}} \{V(A)\}$

Caso base

$i = 0$ $X_0 = \emptyset$ $S_0 = \emptyset$ $V(S_0) = 0$

$i = 1$ $X_1 = \{1\}$ $S_1 = \{1\}$ $V(S_1) = p_1$ → ogni selezione di 1ª sola attività.

Caso passo:

passo ricorsivo > 1 :

S_i → assumendo di avere già risolto sottop. più piccoli $S_0, S_1 \dots S_{i-1}$

prima di continuare → $\boxed{p(i)} = \max \{J \mid J < i$ t.c. J è compatibile con $i\}$

↓

$\max\{\emptyset\} = 0$;

PASSO 1, RISULTA: Se $i \in S_i$ ⇒ (insiemi:) $S_i = S_{p_i} \cup \{i\}$; $V(S_i) =$ (pesi?) $V(S_{p_i}) + V(i)$;

PASSO 2, RISULTA: Se $i \notin S_i$ ⇒ $S_i = S_{i-1}$ → perché i diventa ininfluente sul risultato

$$V(S_i) = V(S_{i-1}).$$

Proprietà della sottostruttura ottima. → guarda nel libro.

primo sottop. comp.

$$S_i = \begin{cases} S_{p(i)} \cup \{i\} \rightarrow \text{se } V(S_{p(i)}) + V_i \geq V(S_{i-1}) \\ S_{i-1} \rightarrow \text{altrimenti} \end{cases}$$

Algoritmo: (ricorsivo, non buono):

```
WIS_R (i)
   if (i = 0): {
       return ∅, 0;        // insieme, valore
   } else: {
       T = WIS_R (p(i));   // {S_p(i) ; V(S_p(i))}  sout. prim. sott. comp.
                           //    T1        T2
       R = WIS_R (S_i-1);  // {S_i-1 ; V(S_i-1)}  sout. sott.
                           //    R1        R2
       if: (T2 + Vi >= R2)
           return (T1 ∪ {i}, T2 + Vi);
       else:
           return (R1, R2);
       }
   }
}
```

→ solo cons. i v.

```
WIS-R-R {
   if i=0 {
       return 0;
   else {
       A = WIS-R-R (p(i))  -> V(S_p(i))
       B = WIS-R-R (i-1)   -> V(i-1)

       if (A + Vi) > B {
           return A+Vi;
       }else{
           return B;
       }
   }
}
```

Soluzione Bottom-up:

Memorizzo V in un array → la var è i di S_i

Risoluz. con la nostra ist. di ese:

$S_{p(i)} + V_i = S_{p(0)} + V_{(2)} = 0 + 2 = 2 > 10$? NO → CASO PASSO 2: $S_i = S_{i-1}$

	0	1	2	3	4	5	6
V	0	10	10	18	18	18	21

$V_{S_6} = 21.$

0 → C.B.

10 → perché $S_{p(i)} + V_i > S_{i-1}$: PASSO 1.
$S_{p(i)} + V_i \rightarrow 0+10$, $S_{i-1} \rightarrow 0$

$S_2 = 10$

$$S_3 = \underbrace{V(S_{p(3)})}_{10} + \underbrace{V_3}_{8} > \underbrace{V(S_2)}_{10}?$$

sì → 18

No → 10

ALGO:

$V[0]=0$ $\quad V[i] \leftarrow V(S_i)$

```
for i=1 to n
    if V[p(i)] + vi >= V[i-1] {
        V[i] = V[p(i)] + vi;
    } else {
        V[i] = V[i-1];
    }
return V[n];
```

S_i: STAMPA DELLA SOLUZIONE.

```
PRINT_WIS(i) :
  if (i ≠ 0):
     if V[p(i)] + vi > Vi-1 {
        print (i);
        PRINT_WIS ( p(i) );
     } else {
        PRINT_WIS (Si-1);
     }
  }
```

$S_i = (S_{p_i} + V(S_i)) > S_{i-1}$

STAMPA:

0	1	2	3 = 4	5	6

$V = \{0, 10, 10, 18, 18, 18, 21\}$

B

CP.1 $S_0 \cup \{10\}$

CP.1 $S_1 \cup \{8\}$

CP.2 $S_4 = S_3$

CP.1 $= S_4 \cup \{6\}$

Manca la dimostrazione → prossima lez.

↓

* PROPRIETÀ DELLA SOTTOSTRUTTURA OTTIMA.

se $i \in S_i \Rightarrow S_i = S_{p_i} \cup \{V_i\}$

se $i \notin S_i \Rightarrow S_i = S_{i-1}$

↓

x PASSARE ALLA RICORR. → aggiungo "vince il migliore"

$$S_i = \begin{cases} S_{p(i)} \cup i & \text{se } \overbrace{S_{p(i)} + V_i \geq V(S_{i-1})}^{\text{"vince il migliore"}} \\ \text{altrimenti} & S_{i-1} \end{cases}$$

PROVA DIMOSTRAZIONE. → HateVille.

- f. comp

HATEVILLE)

n case una di fianco all'altra

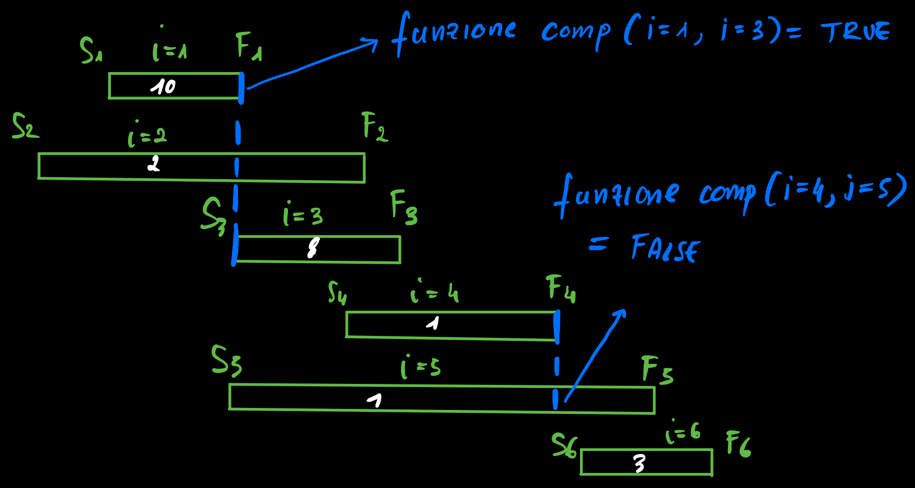
$c_{i-1} \leftarrow c_i \rightarrow c_{i+1}$

c_i = casa, odia i vicini

d_i = quota della c_i

→ sottoinsieme che dà la donaz. massima.

di abitanti che non si odiano.



- = value dell'evento
- = esempio funzione comp
- = evento con tempo start e tempo final

$S_{p(i)} + V_i = S_{p(0)} + V_{(2)} = 0 + 2 = 2 > 10?$ NO

CASO PASSO 2 :

$S_i = S_{i-1}$

V

0	1	2	3	4	5	6
0	10	10	18	18	18	21

$V_{S_6} = 21.$

C.B.

perché $S_{p(i)} + V_i > S_{i-1}$: PASSO 1.

$S_{p(i)} + V_i$ = 0+10

S_{i-1} = 0

COPIA! 2 ∉ S

COMBINA! 3 ∈ S

i-evento :

0	1	$\boxed{2}$	$\boxed{3}$	4	5	
0	10	10	18	18	18	21

CASI BASE