

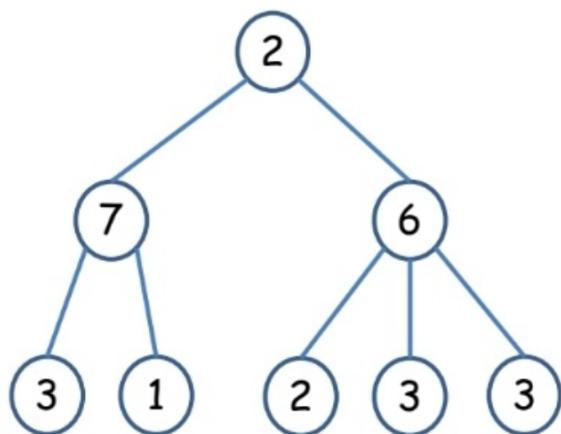
## Esercizio: II di peso massimo su alberi (il problema della festa aziendale)

problema: invita i dipendenti alla festa aziendale

massimizza: il divertimento totale degli invitati

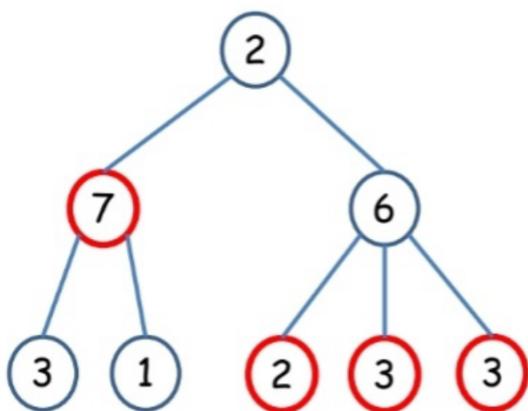
vincolo: tutti devono divertirsi

→ non invitare un dipendente e il suo boss diretto!



input: un albero con pesi sui nodi

goal: un II di peso totale massimo



input: un albero con pesi sui nodi

goal: un II di peso totale massimo

OPT= 15

## PROVA RISOLUZIONE



## SOTTOPROBLEMI

IL SOTTOPROBLEMA J-ESIMO È IL J-ESIMO SOTTOALBERO PER CUI CALCOLO:

- i) II COMPRENDENTE LA RADICE DEL SOTTOALBERO
- ii) II SENZA LA RADICE

## STRUTTURA DATI

CREO UN ALBERO T' DOVE OGNI NODO HA OLTRE CHE I CAMPI VAL E FIGLI I CAMPI:

- i) ROOT → VAL ORC SOLUZIONE OTTIMA COMPRENDENTE LA RADICE
  - ii) DEEP → VAL ORC SOLUZIONE OTTIMA SENZA RADICE
  - iii) MAX → MASSIMO FRA I 2 CAMPI PRECEDENTI
- ↳ USATO PUR : RICOSTRUZIONE È CALCOLO DI DEEP

## CALCOLO SOLUZIONI

ABBIAMO PRINCIPALMENTE 2 CASISTICHE:

- i) V È UNA FOGLIA :

$$V.\text{ROOT} = V.\text{VAL}$$

$$V.\text{DEEP} = 0$$

$$V.\text{MAX} = V.\text{VAL}$$

- ii) V È UN NODO INTERNO :

$$V.\text{ROOT} = V.\text{VAL} + \text{SUM}(V.\text{DEEP})$$

$$V.\text{DEEP} = \text{SUM}(V.\text{MAX})$$

$$V.\text{MAX} = \text{MAX}(V.\text{ROOT}, V.\text{DEEP})$$

→  
VU FIGLIO DI V

## ALGORITMO

UNA VISITA DFS RIC. MODIFICATA IN MODO DA APPLICARE

ALGORITMO ( $v$ ):  $\Rightarrow$  LE OSS FATE PRIMA

IF ( $v$  È FOGLIA):

$V.\text{ROOT} = V.\text{VAL}$

$V.\text{DEEP} = 0$

$V.\text{MAX} = V.\text{VAL}$

RETURN  $V.\text{ROOT}, V.\text{DEEP}, V.\text{MAX}$

FOR EACH ( $u$  FIGLIO DI  $v$ ):  $\Rightarrow$  SARÀ BUC UNA VARIABILE PER FIGLIO

$\text{OPT\_ROOT}_i, \text{OPT\_FIGLI}_i, \text{OPT\_MAX}_i = \text{ALGORITMO}(u)$

$V.\text{ROOT} = V.\text{VAL} + \text{SUM}(\text{OPT\_DEEP})$

$V.\text{FIGLI} = \text{SUM}(\text{OPT\_MAX})$

$V.\text{MAX} = \text{MAX}(V.\text{ROOT}, V.\text{DEEP})$

RETURN  $V.\text{ROOT}, V.\text{DEEP}, V.\text{MAX}$

SOLUZIONE ( $\text{root } v$ ):

CHIAMA CHI EFFETTUÀ LA

$\text{OPT} = \text{ALGORITMO}(v) \Rightarrow$  CHIAMA CHI RICORSIVA

RETURN  $\text{OPT}[3]$

RICOSTRUZIONE\_SIANT ( $v$ ):

$\text{MAX} = \text{SOLUZIONE}(v)$

$\Rightarrow$  CHIAMA LA RICORSIONE SULL'ALBERO USANDO LA SOLUZIONE

IF  $V.\text{ROOT} == \text{MAX}$ :

COME BASE

$\text{RICOSTRUZIONE}(v, \text{TRUE})$

ELSE

$\text{RICOSTRUZIONE}(v, \text{FALSO})$

RICOSTRUZIONE(V, BOOL)

IF ( $V \neq \text{NULL}$ )

RETURN NULL

IF (BOOL == TRUE):

V.MARKE = TRUE

FOR EACH (FIGLIO U DI V)

RICOSTRUZIONE(U, FALSE)

CASE

FOR EACH (FIGLIO U DI V):

IF U.ROOT == MAX: VUOL DIRCI CHE LA ROOT È PRESENTE

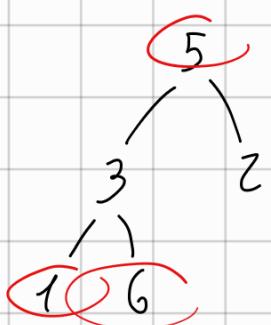
RICOSTRUZIONE(U, TRUE)

ELSE

RICOSTRUZIONE(U, FALSE)

RETURN

VARIETÀ PROBLEMI



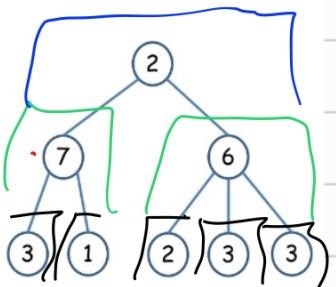
(1)  $\rightarrow 1, 0, 1$

(6)  $\rightarrow 6, 0, 6$

3  $\rightarrow (3, 7, 7)$

2  $\rightarrow (2, 0, 2)$

(5)  $\rightarrow (1, 9, 12) \rightarrow 0 | C$



3  $\rightarrow 3|0|3$

1  $\rightarrow 1|0|1$

7  $\rightarrow 7|4|7$

2  $\rightarrow 2|0|2$

3  $\rightarrow 3|0|3$

3  $\rightarrow 3|0|3$

6  $\rightarrow 6|8|8$

2  $\rightarrow 14|15|(5)$