

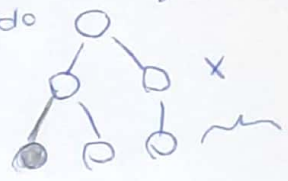
Pregunta 3).

Josefina Nicolet

d) una persona puede estar en la lista de invitados, o no. pero si esta no pueden estar invitados, si es su jefe, su supervisor y sus supervisados. entonces:

La presidenta de la empresa (caso base) puede estar o no invitada

- si esta, entonces el problema se podría resolver buscando la solución óptima desde el tercer nivel de jerarquía, mismo problema pero más pequeño, ya que los supervisados por la presidenta no pueden estar invitados.



- si no esta, entonces el problema se podría resolver buscando la solución óptima desde los supervisados por la presidenta, por lo que es el mismo problema pero con $N-1$ personas.

0 1 0 1 0 1 0 1

b) con $X_i = \{0, 1\}$, 0 si la persona no está invitada y 1 si la persona está invitada a la fiesta y V_i el índice de simpatía. queremos maximizar $\sum_{i=0}^N X_i V_i = \text{OPT}(N)$
 \rightarrow arreglo de las personas

Si la presidenta está invitada tenemos que la solución será $V_1 + \text{Optimo de } N - \{1 \text{ y sus supervisados}\}$
 (1. supervisados)

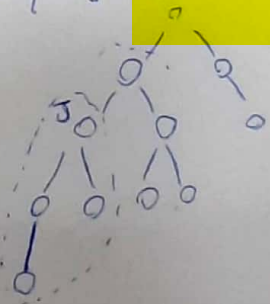
si la presidenta no está invitada tenemos que $\text{Optimo de } N - \{1 \text{ (presidenta)}\}$.

Generalizando, la persona j

asumiendo que se tienen 2 supervisados como máximo y utilizando heap.

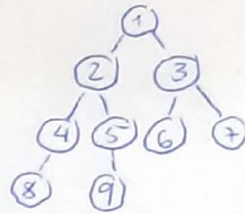
si la persona está invitada, sería igual a encontrar la solución óptima de el árbol hacia arriba ~~hacia~~ sin considerar a mi supervisor y el árbol hacia abajo sin considerar mis supervisados

$$\max \{ V_j + \text{opt}(N - \{j, \text{supervisor} + j \text{ supervisados}\}) \mid \text{opt}(N - j) \}$$



c). si queremos saber la solución optima tenemos que decidir si una persona esta invitada o no, para esto se debe buscar el maximo para lo que necesitamos calcular 2 subproblemas.

e).



~~para saber si A va o no $Opt(1)$ debemos calcular el optimo de $N=2, 3, 4, 5, 6, 7, 8, 9$ si no va y el optimo de $N=4, 5, 6, 7, 8, 9$ si va y ver el Maximo~~
 Pero para calcular el optimo de $N=[2, \dots, 9]$ se debe calcular el de $N=[4, 5, 8, 9]$

Si para saber si 1 va o no ~~en~~ $Opt(1)$ debemos calcular $Opt(2)$ y $Opt(3)$ (si no va) y $Opt(4), Opt(5), Opt(6)$ y $Opt(7)$ si va, para calcular $Opt(2)$ debemos calcular $Opt(4), Opt(5)$ y $Opt(3)$ ^{si va}, para calcular $Opt(3)$ ~~debemos~~ y si no va $Opt(4), Opt(5)$, para calcular $Opt(3)$ debemos calcular $Opt(6)$ $Opt(7)$ si no va y $Opt(2)$ si va, hay una cadena de llamadas dado que hay que optimizar diferentes subarbores, dependiendo de cada decision dado que "se separa el arbol en 2"