Project by:
M. Farrukh Khan (15B-112-SE)
M. AYUB KHAN (15B-120-SE)
MUHAMMAD ALI (15B-077-SE)

PARTY PLAINNING ALGORITHM

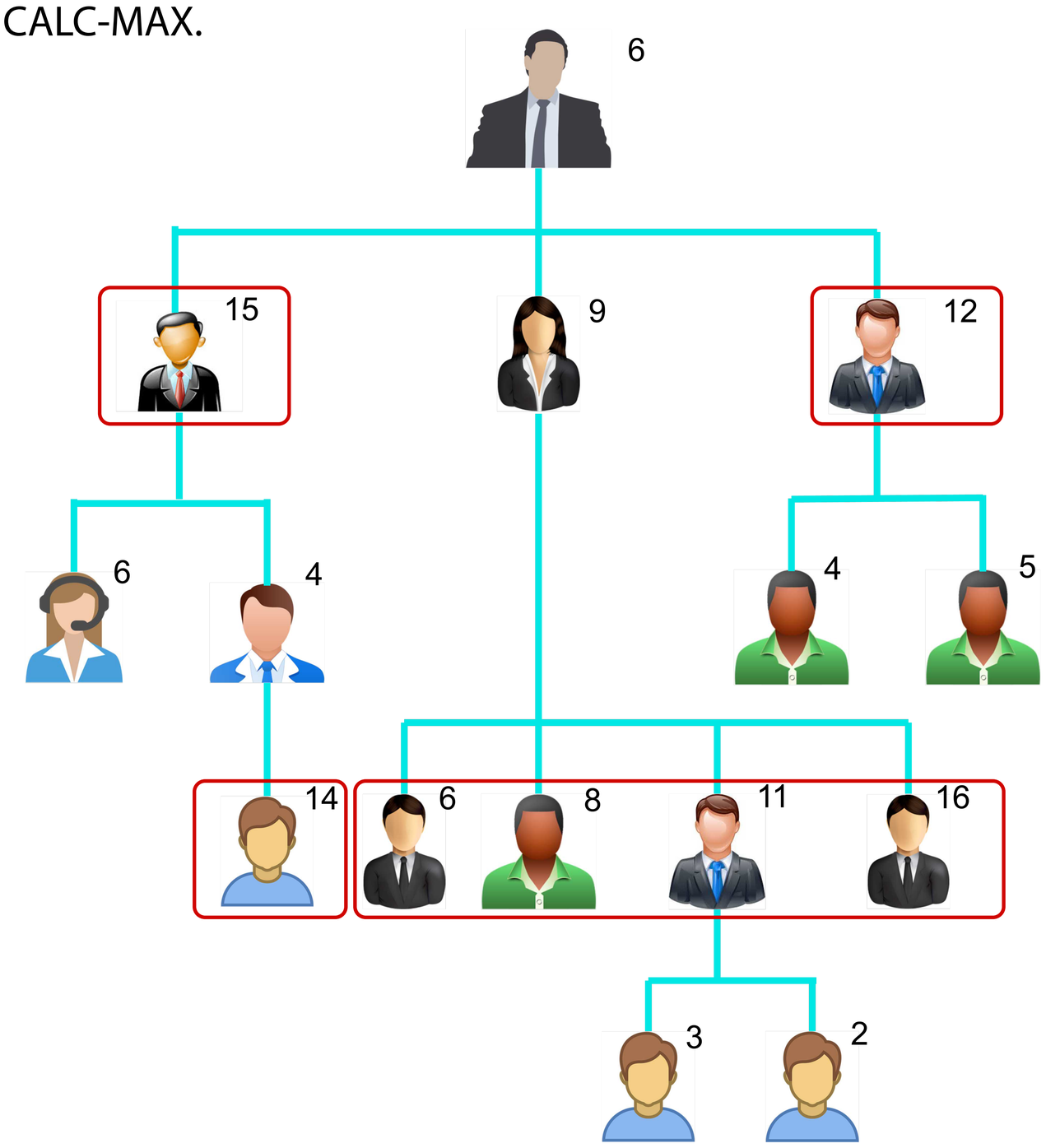
CS322
Design and Analysis of Algorithms

INTRODUCTION

In order to make the party fun for all attendees,
the president doesn't want both an employee & his/her immediate supervisor to attend.
Professor Stewart is consulting for the president of a corporation that is planning a company party.

EXAMPLE

Add three fields to each node: a[i] and p[i] which stores the optimal value in the subtree rooted at i if i is absent and present respectively and guest[i] which indicates whether i is a guest.
Finding the optimal sum can be done by running the following algorithm CALC-MAX from the root and checking whether a[root] or p[root] is larger.
To fill in the guest[i] field, run LIST-GUEST from the root after CALC-MAX.



ALGORITHM

```
CALC-MAX(x)
1 a[x] = 0, p[x] = conviviality[x]
2 y = left-child[x]
3 while y != null
4     Calc-Max(y)
5     y = right-sibling[y]
6 if x is not the root
7     a[parent[x]] = a[parent[x]] + max{a[x], p[x]}
8     p[parent[x]] = p[parent[x]] + a[x]

LIST-GUEST(x)
1 if x = root
2     if p[x] > a[x]
3         guest[x] = true
4     else
5         guest[x] = f alse
6 else
7     if guest[parent[x]] = true
8         guest[x] = f alse
9     else
10        if p[x] > a[x]
11            guest[x] = true
12        else
13            guest[x] = f alse
14  y = left-child[x]
15  while y != null
16        List-Guest(y)
17        y = right-sibling[y]
```

RUN TIME

Since each node is processed once and all operations in the processing run in constant time, the running time is O(n).