Course: CS545
Spring, 2015
Solution for the puzzle
Student: Shuo Yang

Problem

input:

Given n algorithm students $s_1, s_2, \cdots, s_n$ ordered by their ranks from 1 through n, and m chocolate bars. The top-ranked student s_1, proposes a discrete distribution of bars among the students. Each student votes "yes" or "no" for the proposal. If at least half the votes are "yes", the bars are handed out. If the proposal fails, the top student is dismissed and gets no bars, the process repeats among the remaining students (that means, s_2 gets to propose first). Assuming students are making their decisions independently, and smart enough to figure out the optimal way to vote to maximize the number of bars they get, and are extremely competitive, so each votes to dismiss the proposer if they get the same number of bars whether or not the proposal wins.

output:

The optimal distribution proposed by the top student s_1.

Algorithm

Assume that all n students are ranked from 1 to n, 1 is the highest rank and n is the lowest rank. First define some global variables.

$OptDist[1:n]$:= array that holds the optmial distribution, initially , contains all 0s.
So $OptDist[i]$ is the number of bars distributed to the student with the i_{th} rank.
m := total number of chocolate bars
n := total number of students

Function $OptimalBarDistribution$ takes a $rank$ of a student and produces an optimal distribution proposed by that student.

Function $OptimalBarDistribution(rank)$
    if $rank == n$ // base case
        $OptDist[rank] = m$ // assign all bars since no student exists with a lower rank
        return
    $OptDist[rank] := m$ // initialize number of bars to $rank_{th}$ student.
    $next\_rank := rank + 1$
    $OptimalBarDistribution(next\_rank)$
    for $i := next\_rank$ to n
        if $OptDist[i] > 0$
            $OptDist[i] := 0$
        else // $OptDist[i] == 0$
            $OptDist[i] := 1$

$$OptDist[i] := OptDist[i] - 1$$

When the function terminates, $OptDist$ will contain the optimal bar distribution proposed by the $rank_{th}$ student. So to get the optimal bar distribution proposed by the top ranked student, just call $OptimalBarDistribution(1)$.

Correctness

First, suppose among all n students, there are k students remaining (the other $n-k$ stduents dismissed). Let $P(k)$ be: "among all k students, the top ranked (top rank is $n-k+1$) student's optimal bar distribution is to distribute all $0s$ to $\lceil \frac{k-1}{2} \rceil$ students among other students except herself, and to distribute all $1s$ to $\lfloor \frac{k-1}{2} \rfloor$ students among other students except herself, and to distrbute $m - \lfloor \frac{k-1}{2} \rfloor$ bars to herself.

Now we prove that $P(k)$ is true using induction.

Proof. Base case: $P(1)$. When there is only 1 student left, his optimal distribution is to give all m bars to himself because $k - 1 = 0$. So $P(1)$ holds.

Inductive step: Suppose $P(k)$ holds, we need to show that $P(k+1)$ also holds. When there are $k+1$ students left, the number of students with lower rank than the top ranked student is k. Since $P(k)$ is true, the top ranked student will figure out the optimal distribution based on how bars are distributed by the student with the next highest rank. By the induction hypothesis, the next highest ranked student will distribute the bars this way: all $0s$ to $\lceil \frac{k-1}{2} \rceil$ students among other students except himself, and all $1s$ to $\lfloor \frac{k-1}{2} \rfloor$ students among other students except himself, and $m - \lfloor \frac{k-1}{2} \rfloor$ bars to himself. In order to win, the top ranked student must make at least $\lceil \frac{k+1}{2} \rceil$ students vote "yes". Thus, to maximize the number of bars she can get, the top ranked student must distribute in this way: all $1s$ to $\lceil \frac{k-1}{2} \rceil$ students to whom the next highest ranked student distribute 0 bar to, they must vote "yes" to the top ranked student's proposal, otherwise they would end up getting nothing; all $0s$ to $\lfloor \frac{k-1}{2} \rfloor$ students to whom the next highest ranked student distribute 1 bar to; and also 0 to the next highest ranked student, and all the remaining bars to herself. Since each student always votes "yes" to her own proposal, therefore the top ranked student will eventually get total $1 + \lceil \frac{k-1}{2} \rceil = \lceil \frac{k+1}{2} \rceil$ "yes" votes, which are good enough for her to win the proposal. So the top ranked student's optimal distribution would be: all $0s$ to $\lfloor \frac{k-1}{2} \rfloor + 1 = \lfloor \frac{k+1}{2} \rfloor = \lceil \frac{k}{2} \rceil$ students; all $1s$ to $\lceil \frac{k-1}{2} \rceil = \lfloor \frac{k}{2} \rfloor$ students; and all the remaining bars to herself. Thus $P(k+1)$ also holds.

By the principle of induction, $P(k)$ if true for all $n \in \mathbb{Z}^+$. □

Although in order to ensure the top ranked student always wins, this relation must hold:

$$m - \lfloor \frac{n-1}{2} \rfloor \geq 1 \tag{1}$$

When n is odd, we can take off the floor:

$$\begin{aligned} m - \frac{n-1}{2} &\geq 1 \\ m &\geq \frac{n+1}{2} \end{aligned} \tag{2}$$

When n is even, $\lfloor \frac{n-1}{2} \rfloor = \frac{n}{2} - 1$, thus we have:

$$\begin{aligned} m - (\frac{n}{2} - 1) &\geq 1 \\ m &\geq \frac{n}{2} \end{aligned} \tag{3}$$

So we can conclude that:

$$m \geq \lfloor \frac{n+1}{2} \rfloor \tag{4}$$

The correctness of the algorithm follows the proof of $P(k)$ as long as $m \geq \lfloor \frac{n+1}{2} \rfloor$.

Run time analysis

Let $T(n)$ be the run time for the input with n students. The recurrency equation is:

$$\begin{aligned} T(n) &= T(n-1) + \Theta(n-1) \\ T(n-1) &= T(n-2) + \Theta(n-2) \\ T(n-2) &= T(n-3) + \Theta(n-3) \\ &\cdots \\ T(2) &= T(1) + \Theta(1) \\ T(1) &= \Theta(1) \end{aligned} \tag{5}$$

By repeatedly subsitituting the recurrence relation on the right hand side, we have,

$$\begin{aligned} T(n) &= T(1) + \Theta(\sum_{i=1}^{n-1} i) \\ &= \Theta(1) + \Theta(n^2) \\ &= \Theta(n^2) \end{aligned} \tag{6}$$

Therefore the run time is $\Theta(n^2)$.

Answer to the puzzle

In the puzzle, it is the case when $n = 5$ and $m = 100$, the relationship between n and m holds. So by running the algorithm on n and m, we get the optimal distribution proposed by the top ranked student is:

rank-1: 98 bars;
rank-2: 0 bar;
rank-3: 1 bar;
rank-4: 0 bar;
rank-5: 1 bar.