HW5 Solution, CS330 Discrete Structures, Spring 2015

Instructor: Professor Edward M. Reingold
TA: Taeho Jung

April 6, 2015

1 Choosing computer terminal

Let's just consider the case where $n > 3$ so that there are at least two users. If the terminals are maximally used, let's say 'the terminals are packed'.

Every user makes his computer terminal and the adjacent computer terminals unusable. Therefore, the best packing case is when every other computer terminal is used, and the worst packing case is when there are two unused computer terminals between every two users.

It is easy to observe that adding one computer terminal to the worst packing case will increase $U(n)$ by 1 and removing one computer terminal from the best packing case will decrease $U(n)$ by 1. Therefore, to derive the function $U(n)$, we need to first figure out when the computer terminals are best packing and worst packing.

To be the best packing case, every time a user chooses a computer terminal, it must be the exact middle point between two used terminals (otherwise the computer terminals can be more packed by removing several terminals, which can be proved by contradiction), and finally there is one empty terminal between every two users when the terminals are packed. Then, $n-1$ must be a perfect power of 2 (*i.e.,* $n-1=2^k$ for some integer k) because every user finds exact middle point, which means the distance between two users at any choice must be divisible by 2 (distance between the users is defined as empty terminals between two users +1). This means when $n = 2^k + 1$ for some k, the terminals are best packed.

Similarly, to be the worst packing case, every time a user chooses a terminal, it must also be the exact middle point (similar contradiction can show that otherwise one can add some terminals to make it less packed). When the terminals are finally packed, each user has two unused terminals at his right side except the last user, which used to be just one terminal in the best packing case. That is, except the last user, every user occupies 3 terminals now while 2 were occupied per user in the best case. Therefore, if we remove the right-most terminal from the best case (terminal #: $n-1=2^k$), add one unused terminal at the right side of every user (terminal #: $\frac{3}{2}\cdot 2^k$) and add back the right-most terminal (terminal #: $2^k \cdot \frac{3}{2} + 1$), the best packed case becomes the worst packed case. Therefore, when $n = \frac{3}{2}\cdot 2^k + 1$ for some k, terminals are worst packed.

$$\text{Terminals are } \begin{cases} \text{best packed, when} & \exists k : n = 2^k + 1 \\ \text{worst packed, when} & \exists k : n = \dfrac{3}{2}\cdot 2^k + 1 \end{cases}$$

Obviously, $U(n)$ at the best/worst cases are:

$$U(n) = \begin{cases} \frac{n-1}{2} + 1 = \dfrac{n+1}{2} = 2^{k-1} + 1, & \text{when best packed} \\ \frac{n-1}{3} + 1 = \dfrac{n+2}{3} = 2^{k-1} + 1, & \text{when worst packed} \end{cases}$$

Then, we have the following for $U(n)$:

$$\forall k > 0 : U(n) = 2^{k-1} + 1, 2^k + 1 \leq n \leq \frac{3}{2} \cdot 2^k + 1$$

However, n can fall into other interval besides $2^k + 1 \leq n \leq \frac{3}{2} \cdot 2^k + 1$, and let's consider the case when $\frac{3}{2} \cdot 2^k + 1 < n < 2^{k+1} + 1$. Remember that adding one terminal to the worst case will increase the $U(n)$ by 1. In fact, this is true because one of the two empty terminals between the users will become:

$$Used \;\; Empty \;\; Empty \;\; Used \Rightarrow Used \;\; Empty \;\; Used \;\; Empty \;\; Used$$

In addition, it can be proved by contradiction that until n gets to the next best case $n = 2^{k+1} + 1$, $U(n)$ gets increased by 1 every time we add one more terminal. Therefore, the final formula of $U(n)$ is:

$$\forall k > 0 : U(n) = \begin{cases} 2^{k-1} + 1, & 2^k + 1 \leq n \leq \frac{3}{2} \cdot 2^k + 1 \\ n - 2^k, & \frac{3}{2} \cdot 2^k + 1 < n < 2^{k+1} + 1 \end{cases}$$

2 Exam problem sharing

(a)

Suppose students are numbered $(1,2,3,\cdots,n)$. Student 1 sends his problem to student 2, student 2 sends his problem along with received one to student 3, and this is repeated until all problems reach student n.

(b)

In order to gather all problems, $n-1$ students needs to at least send out one message. After that, in order to disseminate the gathered problems, at least $n-1$ messages are needed for $n-1$ students. That is, theoretic lower bound of the required messages is $2n-2$, which is exactly the number of messages sent out in the previous greedy algorithm. Therefore, the algorithm does minimize the number of messages.