CS 1510 Homework 4

Brian Falkenstein, Brian Knotten, Brett Schreiber

September 4, 2018

9

a

This algorithm can be proven incorrect by considering a situation where large files, which by the algorithm will be placed after shorter files, have a higher probability of being accessed. This is demonstrated most simply in an example with 2 files, F_1 of size 2 and probability 0.1, and F_2 of size 10 and probability 0.9. Following the algorithm, F_2 would be placed after F_1, giving an expected access time of $(2 * 0.2) + (12 * 0.9) = 11$. This can be improved by simply swapping the locations of the two files, giving us an expected access time of $(10 * 0.9) + (12 * .1) = 10.2$.

b

Assume the algorithm stated in the problem, A, is incorrect and has some input I that causes it to give the incorrect output. Define $Opt(I)$ to be an acceptable output to the problem that agrees with $A(I)$ for the most steps. That is, before their first point of disagreement, $Opt(I)$ and $A(I)$ chose the same ordering of files. At this point of disagreement, call the file that $A(I)$ chose F_j and the file that $Opt(I)$ chose F_k. We know that F_k must have a lower probability than F_j, otherwise $A(I)$ would've already chosen it. That leaves us with two scenarios:

1. The length of F_j is greater than that of F_k.
2. The length of F_j is less than that of F_k.

Both situations bring us to a contradiction of $Opt(I)$ being an acceptable output. Having the larger probability file come before the lower probability one always results in a more optimal solution, as the probability for a specific file will always stay the same no matter where it is placed in the chain, but the length of files before the file, and thus the value multiplied by the probability at that step in the calculation of expected access time, varies. The highest probability files should be placed at the start of the tape, so as to minimize the length of tape before these files that will need to be multiplied by their probabilities.Thus, $Opt(I)$ can be modified to agree with $A(I)$ for one more step by simply swapping F_k with F_j, making $Opt(I)$ more optimal.

c

Assume to reach a contradiction that the presented algorithm, A, is incorrect. Therefore there exists an input I that causes A to give a suboptimal output. Let $OPT(I)$ be the optimal solution to the problem that agrees with $A(I)$ for the most number of steps i.e. up to the first disagreement between $A(I)$ and $OPT(I)$, $A(I)$ and $OPT(I)$ have the same ordering of the files. Because $A(I)$ sorts the files in increasing order of the ratio of length over access probability, it must be the case that at the first point of disagreement there exists files F_m, F_n where $\frac{\ell_m}{p_m} > \frac{\ell_n}{p_n}$ but F_m is listed before F_n in $OPT(I)$ and F_n is listed before F_m in $A(I)$. Because F_n and F_m are adjacent in the the list, exchanging them will only effect their respective access times. Therefore, the expected access time before exchanging F_n and F_m is (where $\ell_{pre} = \sum_{i=1}^{m-1} \ell_i$):

$$p_m(\ell_{pre} + \ell_m) + p_n(\ell_{pre} + \ell_m + \ell_n)$$

and the expected access time after the exchange is:

$$p_n(\ell_{pre} + \ell_n) + p_m(\ell_{pre} + \ell_n + \ell_m)$$

If exchanging F_m and F_n reduces the overall access time, then $OPT(I)$ can be modified into $OPT'(I)$ where $OPT'(I)$ agrees with $A(I)$ for one more step than $OPT(I)$, contradicting our statement that $OPT(I)$ agrees with $A(I)$ for the most number of steps. Then, $A(I)$ is correct.

If swapping F_m and F_n does not reduce the overall access time, then we have:

$p_m(\ell_{pre}+\ell_m)+p_n(\ell_{pre}+\ell_m+\ell_n) \leq p_n(\ell_{pre}+\ell_n)+p_m(\ell_{pre}+\ell_n+\ell_m)$

By distributing and cancelling $p_m\ell_{pre}$, $p_m\ell_m$, $p_n\ell_{pre}$, and $p_n\ell_n$ on both sides, we get:

$p_n\ell_m \leq p_m\ell_n = \frac{\ell_m}{p_m} \leq \frac{\ell_n}{p_n}$

Therefore it cannot be the case that $(\frac{\ell_m}{p_m} > \frac{\ell_n}{p_n})$ as stated earlier. Therefore, $OPT(I)$ can be modified into $OPT'(I)$ where $OPT'(I)$ agrees with $A(I)$ for one more step than $OPT(I)$, contradicting our statement that $OPT(I)$ agrees with $A(I)$ for the most number of steps. Therefore, $A(I)$ is correct.

10

a

This algorithm A is not correct, because it does not find an optimal solution for the followind input I:
$p = (3, 6, 10)$
$s = (1, 4, 8)$
$A(I)$ outputs:

1. The first smallest difference is $|p_1 - s_2| = |3 - 4| = 1$. So p_1 gets assigned s_2.
2. The next smallest difference is $|p_2 - s_3| = |6 - 8| = 2$. So p_2 gets assigned s_3.
3. p_3 gets assigned s_1 by default. $|p_3 - s_1| = |10 - 1| = 9$.

$A(I)$'s sum of differences in height is $1 + 2 + 9 = 12$.

There exists a more optimal output $Opt(I)$:

1. p_1 gets assigned s_1. $|p_1 - s_1| = |3 - 1| = 2$.
2. p_2 gets assigned s_2. $|p_2 - s_2| = |6 - 4| = 2$.
3. p_3 gets assigned s_3. $|p_3 - s_3| = |10 - 8| = 2$.

$Opt(I)$'s sum of differences in heights is $2 + 2 + 2 = 6$. So $A(I)$ is not a correct algorithm.

b

Assume for contradiction A is not correct. Then there exists an input I in which $A(I)$ produces a greater total difference in heights than an optimal output $Opt(I)$. Let $Opt(I)$ be the optimal output which agrees with $A(I)$ at the greatest number of steps.

There exists some assignment for p_i where $A(I)$ and $Opt(I)$ first disagree. $A(I)$ assigned ski a, but $Opt(I)$ assigned ski b. $b > a$, since there are no shorter skis that $Opt(I)$ could have assigned, since $A(I)$ and $Opt(I)$ have agreed that the shortest skier gets the shortest ski up to this point. Moreover, $Opt(I)$ must assign a to a taller skier $s_j > s_i$.

For $Opt(I)$ to be more optimal than $A(I)$, $|p_i - b| + |p_j - a| < |p_i - a| + |p_j + b|$, but this is not the case since $p_i < p_j$ and $a < b$. There is a contradiction, and so A is a correct algorithm.