Homework 4

Adam Karl

September 29, 2020

1 Problem 19 (6 points)

1.1 One-Page Solution

Given an example input

time	1	2	3	4	5	6
page A	1	3	5	3	4	-

Add 3 rows:

- requests improved by adding a broadcast here
  - aggregate sum of requests before this point, but at/after the previous broadcast
  - 0 when there is a broadcast here
- time to next broadcast
  - equal to the decrease in wait time for each request this broadcast would help
- improvement by adding a broadcast here
  - the decrease in total wait time
  - found by multiplying the two rows above

A mandatory broadcast must be made right after the last request in order for a feasible solution to be possible. Then, the table looks like:

time	1	2	3	4	5	6
page A	1	3	5	3	4	-
requests improved by adding a broadcast here	-	1	4	9	12	0
time to next broadcast	5	4	3	2	1	0
improvement by adding broadcast here	-	4	12	18	12	0

The optimal place to insert a broadcast is at the maximum value in the bottom row. Here, that is at t=4 to decrease the total wait time by 18.

After adding a broadcast at t=4 and updating the bottom 3 rows the table now looks like:

Now the maximum improvement can be found by inserting the next broadcast at t=3 to decrease the total wait time by 4.

This algorithm takes O(t) time to update the table after each broadcast (update row 3 to the right of the inserted broadcast, update row 4 to the left of it, and for any updated cell recalculate row 5). To add k broadcasts, it runs in O(tk).

This process can be continued as necessary for k broadcasts (as long as $k < t$ otherwise there will be broadcasts inserted that effectively do nothing)

time	1	2	3	4	5	6
page A	1	3	5	3	4	-
requests improved by adding a broadcast here	-	1	4	0	3	0
time to next broadcast	3	2	1	0	1	0
improvement by adding broadcast here	-	2	4	0	3	0

1.2 Multi-Page Solution

It's actually extremely simple to go from a single-page solution to a multi-page solution. Each page in the multi-page solution will have their own table that we will keep updated. We start giving each page a broadcast immediately after the last request for that page. Note that if k is less than the number of pages, there is no feasible solution. Then, while the total broadcasts is less than k, consider how much adding a broadcast to each page can would improve the total waiting time (the maximum of the last row for any page). Add the broadcast that maximizes this improvement (choosing arbitrarily in the event of a tie) and update the page's table. When updating the table, a new best location for a broadcast and smaller

Since it still takes t time to update after inserting a broadcast (no matter which page gets a broadcast), the total runtime is still O(tk).

Extracting the solution: Consider the 4rd row in the table for each page (time to next broadcast). A 0 indicates a broadcast at this time.