Homework 15

Brian Knotten, Brett Schreiber, Brian Falkenstein

September 30, 2018

23

This problem is a variation of the set partition problem where we are partitioning a set of size n (the n request times) into k subsets (each subset is the collection of requests satisfied if the information is sent at one of the k broadcasts). There are a max of $\binom{n-1}{k-1}$ possible partitions: there are n possible times for each of the k broadcasts (time 2 to time $n+1$), but the final broadcast must go after the last nonzero request time, so there are a max of $n-1$ possible times for each of the $k-1$ broadcasts.
Let B be the list of k broadcasts and let m be the time of the last nonzero request.The algorithm starts by placing each of the k broadcasts at the farthest feasible time i.e.:
for broadcast $i = 1$ to k:
    place partition i at slot m+2-i
so that the first (assigned) broadcast is at time $m+1$, the second broadcast is at time m, etc.
Then: let $minSum = \infty$ and let $bTimes =$ the current positions of the k broadcasts.
for broadcasts $i = k$ to 1:
    let ℓ be the number of broadcasts sent thus far (i.e. if $i = k$, one sent if $i = k-1$, two sent, etc.)
    for each $\binom{n-i+1}{\ell}$ possible broadcast time:
        if $totalWaitTime \leq minSum$:
            $minSum = totalWaitTime$
            $bTimes =$ current positions of the k broadcasts
The first for loop iterates over the k broadcasts. The second for loop iterates over the k broadcasts, and for each loop it considers a max of $\binom{n}{k}$ combinations. Note that due to the symmetry of binomial coefficients, $\binom{n}{k} = \binom{n}{n-k}$. Therefore $\binom{n}{k}$ reaches is max value when $k = \lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil$. Therefore the algorithm is $O(min(n^k, n^{n-k}))$

26