Ali Mirabzadeh
305179067
CS 180 – HW3
3.

1.Let's call the optimal set, O.
2.Let's call our algorithm, A.
3.For all indices $r \leq k$ we have $f(i_r) \leq f(j_r)$.
we will simply show
that $|A| = |O|$, that is, that A contains the same number of trucks as O and
hence is also an optimal solution.
Let $i_1, \ldots, i_k$ be the set of boxes in A in the order they were added to A. Note that $|A| = k$.
Similarly, let the set of boxes in O be denoted by $j_1, \ldots, j_m$. Our goal is to prove that $k = m$.
I prove this by Contradiction.
Hence, we assume that A is not optimal, then an optimal set O must have more requests, that is, we must have $m > k$.
Applying 3 with $r = k$, we get that $f(i_k) \leq f(j_k)$. Since $m > k$, there is a
request j_{K+1} in O. This request starts after request j_k ends, and hence after
i_k ends. So, after deleting all trucks that are not compatible with trucks
$i_1, \ldots, i_k$, the set of possible trucks R still contains j_{K+1}. But the greedy
algorithm stops with request i_K, and it is only supposed to stop when R is
empty—a contradiction.

Ali Mirabzadeh
305179067
CS 180 – HW3
7.
First sort the tasks based on p_i, as in the shortest time be the first one to feed into the supercomputer and the longest one be the last one to be fed into the supercomputer
Then
For each job in the sorted array, $J_{1}, \ldots, J_n$
    If P_i is over
        Then pass this job to a PC
    endIF
    increment the job in the array, J++
endfor
This is O(n) which is polynomial

Ali Mirabzadeh
305179067
CS 180 – HW3
12.
a. FALSE,
Let's have (b1, t1) = (500, 1), (b2, t2) = (1000, 1) and r=800
Here, the combination of two is less then rt: 500+1000 < 800*2
But the second stream itself is larger than r: 1000>800
b.
The algorithm should check if the b_{total}<r* t_{total}
First, Sort the streams based on their bits from the shortest to the longest.
For each stream, $b_{1,\dots,}b_n$

```
    bitTotal= bitsTotal + b_i
    if bitTotal > r
        then return false // didn't meet the requirement
    endif
endfor
return true // there is a valid schedule
```

Ali Mirabzadeh
305179067
CS 180 – HW3
16.
First sort times of Xs in non-decreasing order and sort other intervals by ending times
Then For each time in x
    Match each x with the earliest ending unmatched interval.
    If that was possible and $x_i > t_i - e_i$
        Then we have our match, return true
    Endif
Endfor
Return false, nothing found