(2) a) counter example: Taking the activity with least time

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	4	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

The greedy solution will be (2, 8, 11) but which is not optimal one. (1, 4, 8, 11) is the optimal solution.

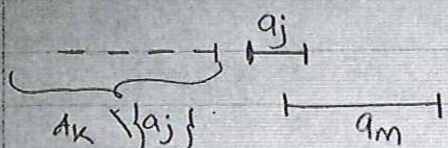
b) Sort the activities in decreasing order on starting time.

```
GreedySelection (s,f) {
        n ← length (s) ;
        A ← {a1} ;          // last started activity
        k ← 1 ;
        for  m = 2 to n
             if  f(m) <= S(k)
                 A = A ∪ {am} ;
                 k = m ;
        return A ;
}
```

Theorem: Consider any non empty subproblem S_K and let a_m denote the activity in S_K with the last activity to start. Then a_m is included in some maximum size subset of compatible activities of S_K

To prove that our algorithm is correct we need to prove the above theorem.

proof: let A_K be an optimal solution
and let a_j be the activity in A_K with the
last activity to start. If $a_j = a_m$ we are done
So assume $a_j \neq a_K$
Now consider the set $A_K' = A_K \setminus \{a_j\} \cup \{a_m\}$



Here all the activity in A_K has finish time earlier than the start time of a_j and the start time of a_m is later then the start time of a_j. So A_K' is feasible and $|A_K'| = |A_K|$.
So A_K' is optimal and algorithm is correct.

③ // House[] contains sorted distance from the begining of the line.

```
Cell Phone Tower ( House [] ) {
1.      Tower = 0,  D = 0;
2.      Find a H in House[] > D
3.      put the tower at H+4,   T = T+1;
4.      D = H+8;
5.      Repeat 2 to 4 until any H in House[]
                                      left.
}
```

proof of correctness:

To prove that this algorithm gives the smallest number of towers T. The current algorithm starts from the first house and covers all subsequent houses within 8 miles with the first tower. Then the algorithm starts with the next house which is not covered yet. The algorithm terminates when there is no house left to cover.

Now lets assume T is the array of towers placed using the above algorithm. If there is an optimal solution T´ that contains less tower that is |T| > |T´| than there will be at least one tower t_i in T that can be removed and still all the houses will be covered. It is possible only if —

1) There is no house in the coverage area of t_i or
2) The houses are already covered by other towers.

Option 1) is not possible as we have at least 1 house from which we place a tower after 4 mile.

Option 2) is also not possible as we are considering to place a tower after the coverage distance of the previous one.

So there is no T´ for which |T| > |T´|

Thus T is the optimal solution.

Running time of the algorithm is O(n)