The point is, ladies and gentleman, greed is good. Greed works, greed is right. Greed clarifies, cuts through, and captures the essence of the evolutionary spirit. Greed in all its forms, greed for life, money, love, knowledge has marked the upward surge in mankind.

— Michael Douglas as Gordon Gekko, 
  Wall Street 
  (1987)
General marking philosophy: a student who gives enough of an answer to show they understood what they were supposed to do, even if they couldn’t do it (or made lots of errors while doing it) should get at least 50% on that question.

Full marks should be given if a solution is sound and not missing anything important.

Feel free to give marks like 14.5/15 to a solution that is correct but contains a minor error.

A student should only get 0 on a question if they made no attempt to answer it at all.
Question 1 (30 Marks)

You have won the contract to install Wi-Fi nodes along a very straight and sparsely populated stretch of road which runs due east and west across the tiny nation of Occiput. There are N houses along the road – each house is identified by its distance from the east end of the road. Each house is located right on the road, not set back from the road. Your assignment is to install Wi-Fi nodes along the road so that each house is no more than 1 kilometre from a node. You can install nodes anywhere along the road – the nodes do not have to be located at houses. You want to install as few nodes as possible.

This figure illustrates an instance of the problem and one possible solution. The black dots represent houses, the white dots represent Wi-Fi nodes, and the grey bars show the “1 km in each direction” range of each Wi-Fi node. The solution shown is not optimal.

(a) (10 marks) Give a Greedy Algorithm to find an optimal (minimal) set of locations for the Wi-Fi nodes. (Hint: consider the west-most house – how far east of that house can you place the first node?)

sort the houses in west-to-east order
while at least one house is not covered:
    let x be the west-most uncovered house
    place a Wi-Fi node exactly 1 km east of house x
or equivalently:

sort the houses in west-to-east order
for h in the sorted list of houses:
    if h is not covered by a previously placed Wi-Fi node:
        place a Wi-Fi node exactly 1 km east of h

Marking: the algorithm can be run from east-to-west without affecting its correctness.

Deduct 1 mark if the student forgets to sort the houses.

For algorithms that are greedy but do not find an optimal solution (for example, “place the first node where it can cover the most houses”) give about 7 marks. For algorithms that aren’t really greedy (for example “place a Wi-Fi node right on every house”) give about 5
marks. For algorithms that do not find a feasible solution, give about 4 marks.
b) **(10 marks)** Prove that the first choice your algorithm makes for a node location is correct (i.e. that there is an optimal solution that contains this location as its first location).

The algorithm’s first choice is to place a node 1 km east of the west-most house. Call this location $a_1$.

Let $O$ be an optimal solution, and let $o_1$ be the west-most node in $O$. $o_1$ cannot be east of $a_1$, since then the west-most house would not be covered by any node in $O$. Thus either $o_1 = a_1$, or $o_1$ is west of $a_1$. If $o_1 = a_1$ then $a_1$ is contained in an optimal solution. If $o_1$ is west of $a_1$, then a node at $a_1$ will cover all the houses that a node at $o_1$ covers. Thus we can remove $o_1$ from $O$ and replace it with $a_1$. This gives a feasible solution with the same cardinality as $O$, i.e an optimal solution that contains $a_1$.

Thus there is an optimal solution that contains $a_1$.

**Marking:** The key idea here is that the algorithm’s first choice can be substituted into any optimal solution that doesn’t already contain it. If the student has that idea, they should get at least 6 marks, even if they couldn’t come up with a proof.

c) **(10 marks)** Complete the proof that your algorithm finds an optimal solution.

Clearly if there is only 1 house, any optimal solution contains one node. The algorithm finds an optimal solution in this base case.

Assume the algorithm finds an optimal solution when there are $\leq n$ houses.

Suppose there are $n+1$ houses. Let $A = \{a_1, a_2, \ldots, a_s\}$ be the algorithm’s solution, and let $O = \{a_1, o_2, o_3, \ldots, o_t\}$ be an optimal solution, in west-to-east order, containing $a_1$ (we know that such a solution exists). We need to show $|A| = |O|$.

By our inductive assumption, $\{a_2, \ldots, a_s\}$ is an optimal solution to the problem of covering all the houses not covered by $a_1$. But this is exactly the same problem that is solved by $\{o_2, \ldots, o_t\}$. Therefore $|\{a_2, \ldots, a_s\}| \leq |\{o_2, \ldots, o_t\}|$. Therefore $|A| \leq |O|$. $|A| < |O|$ is impossible since $O$ is optimal. Therefore $|A| = |O|$, so $A$ is optimal too.
Marking part c): Induction is a very natural way to prove this. The base case is worth 3 marks, and the inductive part is worth 7. If they have the basic idea of induction but don't give a sound proof, they should still get at least 6 marks.

An alternative, non-inductive proof might look like this:

Let $A = \{a_1, a_2, \ldots, a_s\}$ be the algorithm's solution, and let $O = \{o_1, o_2, \ldots, o_t\}$ be an optimal solution. Using the argument already given, we can see that $O' = \{a_1, o_2, \ldots, o_t\}$ is also an optimal solution. Now we can make a similar argument that $a_2$ can be used to replace $o_2$, giving $O'' = \{a_1, a_2, o_3, \ldots, o_t\}$ is a feasible solution with the same cardinality as $O$, so $O''$ is also optimal. Repeating this argument, we replace all the $o$'s with $a$'s, always maintaining optimality. We end up with $A$ being optimal.
Question 2 (20 Marks)

You have landed a prestigious new job, hiring guards for the National Prison for Disgraced Politicians (a very crowded place). The prisoners must be guarded from 6 AM to 6 PM. There are a total of \( n \) guards, but each guard is only available for a specific time period during the day: Guard \( G_i \) will work during the interval \([s_i, f_i]\), where \( 0 \leq s_i < f_i \leq 24 \). Each guard is payed the same amount, regardless of how long their shift is. Since you are paying them out of your own salary, your goal is to hire as few guards as possible.

You may assume that there is a feasible solution – there are enough guards to cover the whole day.

(a) (10 marks) Give a Greedy Algorithm to find an optimal solution (i.e. minimal number of guards) subject to the constraint that there must be at least one guard on duty at all times between 6 AM and 6 PM. The total time period covered may start before 6 AM and may end after 6 PM.

In pseudo-code:

Sort the guards by their start times (earliest first)
\( Time\_covered = S-1 \)
\( index = 1 \)
while \( Time\_covered < F \):
    \( best\_guard = nil \)
    \( best\_guard\_end = 0 \)
    while \( s_{index} \leq Time\_Covered + 1 \):
        if \( f_{index} > best\_guard\_end \):
            \( best\_guard = index \)
            \( best\_guard\_end = f_{index} \)
        \( index ++ \)
    hire guard \( G_{best\_guard} \)  # ie add \( G_{best\_guard} \) to the solution
\( Time\_covered = best\_guard\_end \)

in English:
Sort the guards by their start times (earliest first).
From the guards that cover \( S \), choose the one with the latest finish time. Continue with that time + 1 as the new start time.
We never have to go back and look at a guard twice because we only reject a guard if we have found a better one (i.e., one who covers the same required start time and whose end-time is later).

Marking: Similar to Question 1 (a). The algorithm can be presented descriptively or in code or pseudo-code.

Students are not required to give any justification for their algorithm. I included the “never have to go back” comment for the benefit of the reader.

Note: students may have interpreted the question to mean that when guards relieve each other, they must overlap (e.g., if the first guard ends her shift at time x, then the second must start no later than time x-1). This is a reasonable interpretation and should not be penalized. It doesn't affect the structure of the algorithm, just the criterion for deciding if a guard can feasibly be added to the solution.
(b) **(10 marks)** Explain why your algorithm would not work if there is an added constraint that each guard has a first name (Kim, Pat, Kelly, etc) and you cannot hire two guards with the same first name.

Suppose there are two guards named Kim – call them Kim1 and Kim2. The algorithm’s first choice might be Kim1, and then on a later iteration, the best – or perhaps the only – choice might be Kim2. If the algorithm chooses Kim2, it violates the constraint – if it doesn’t choose Kim2, it may not find a solution at all.

Thus we can construct an instance where the algorithm fails.

Marking: the key idea is that for a greedy algorithm to be successful, its choices should be based on purely “local” information. It should not be the case that the optimal first choice needs to consider future optimal – or essential – choices.

Students can explain this clearly or give an example for full marks.

An alternative (and fully acceptable) demonstration of the failure would be to show that proof by induction would not be possible. We can assume that the algorithm makes an optimal first choice and that it finds an optimal solution to the reduced problem after the first guard is chosen, but that does not guarantee that the algorithm’s first choice and the optimal solution to the reduced problem can be combined – as described above, the constraint on names might be violated.

Give part marks in the range 7 to 10 for answers that come close to giving a good explanation or an example of how the algorithm could fail.

Give marks of 5 or less for answers that show some understanding but cannot identify (in any clear way) how the algorithm might fail.