

COMS20010 — 2020–21 Exam

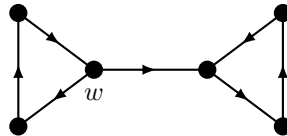
This exam was actually given as a Blackboard test in a slightly different format to the one used this year (hence “Section 1” and “Section 2”). This means this isn’t quite the final version, and since it hadn’t gone through the final round of checking at that point there might still be one or two errors lurking in the model answers, but it should be a good guide as to what to expect.

Section 1

1. (5 marks) (Short question.) Mark each of the following statements true or false:

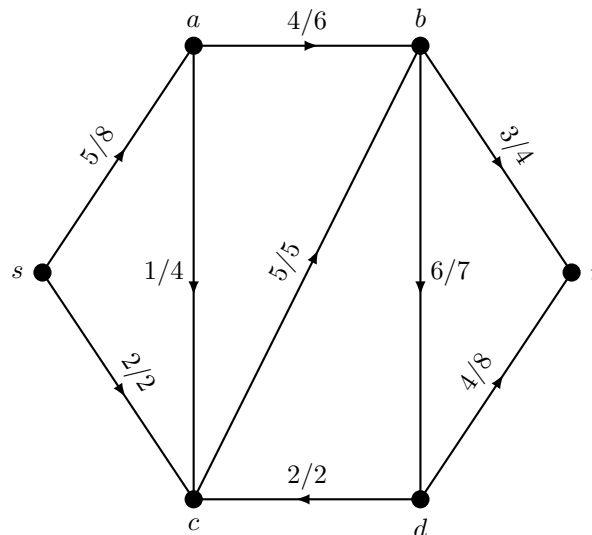
- (a) (1 mark) $n^2 \in O(5n)$.
- (b) (1 mark) $10n^2 + n \in O(n^2)$.
- (c) (1 mark) $\log n \in O(n^{1/5})$.
- (d) (1 mark) $n^{65} \in O(n!)$.
- (e) (1 mark) $2^{n^2} \in O(2^n)$.

2. (5 marks) (Short question.) Consider the following graph G and the indicated vertex w .



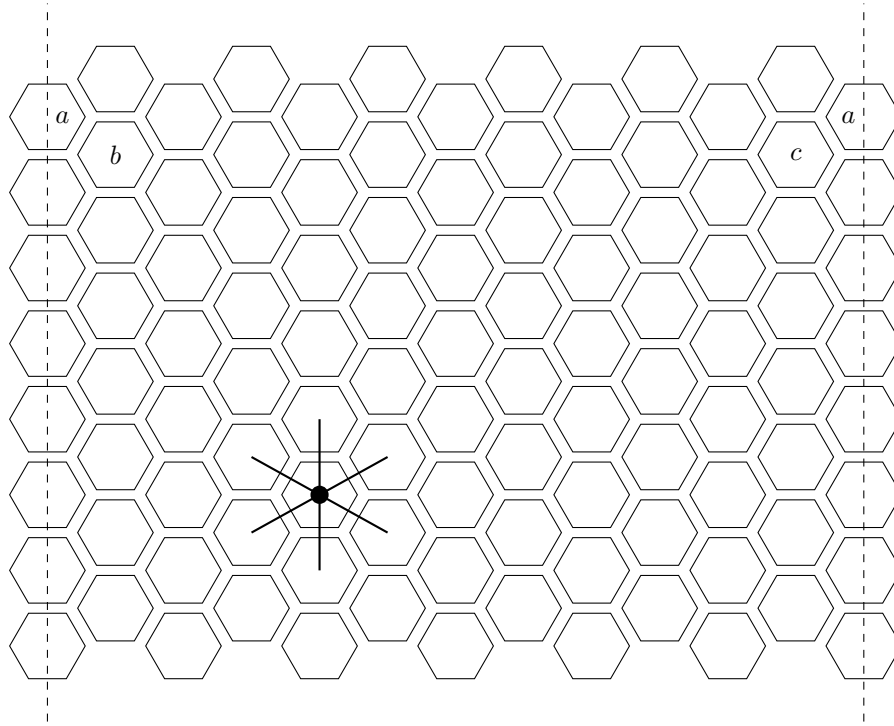
Mark each of the following statements true or false:

- (a) (1 mark) G is weakly connected.
 - (b) (1 mark) G is strongly connected.
 - (c) (1 mark) G has an Euler walk.
 - (d) (1 mark) G contains a directed cycle as an induced subgraph.
 - (e) (1 mark) $d^-(w) = 2$.
3. (5 marks) (Short question.) Consider the following flow network and flow, with source s and sink t .



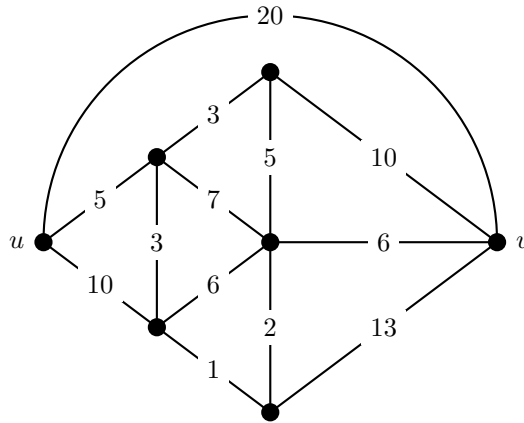
List all augmenting paths of the network under the given flow.

4. (5 marks) (Short question.) The strategy game Civilization VI is played on a hexagonal map as shown below. The map is effectively a cylinder, with the left and right boundaries are joined together, and the top and bottom boundaries are impassable — thus in the picture below, tile *a* appears twice, and is adjacent to both tile *b* and tile *c*. Units can move between any pair of adjacent tiles, and this is stored in memory as a graph in which there is an edge between two tiles if they are adjacent in the map.



If the map is 90 tiles wide and 56 tiles high, how many edges does the graph have? (**Hint:** Use the handshaking lemma.)

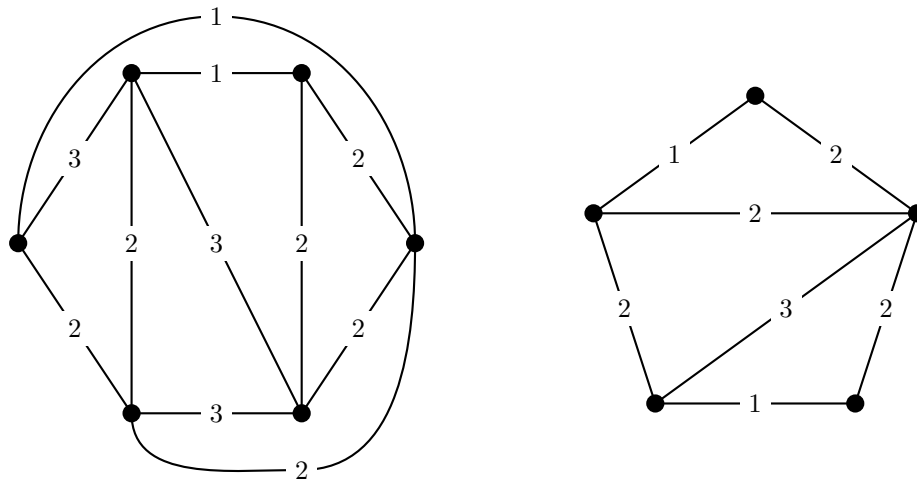
- A. 13440.
 - B. 14829.
 - C. 14940.
 - D. 29658.
 - E. None of the above.
5. (5 marks) (Short question.) Consider the weighted graph below.



What is the distance from u to v ?

- A. 1.
- B. 16.
- C. 18.
- D. 20.
- E. None of the above.

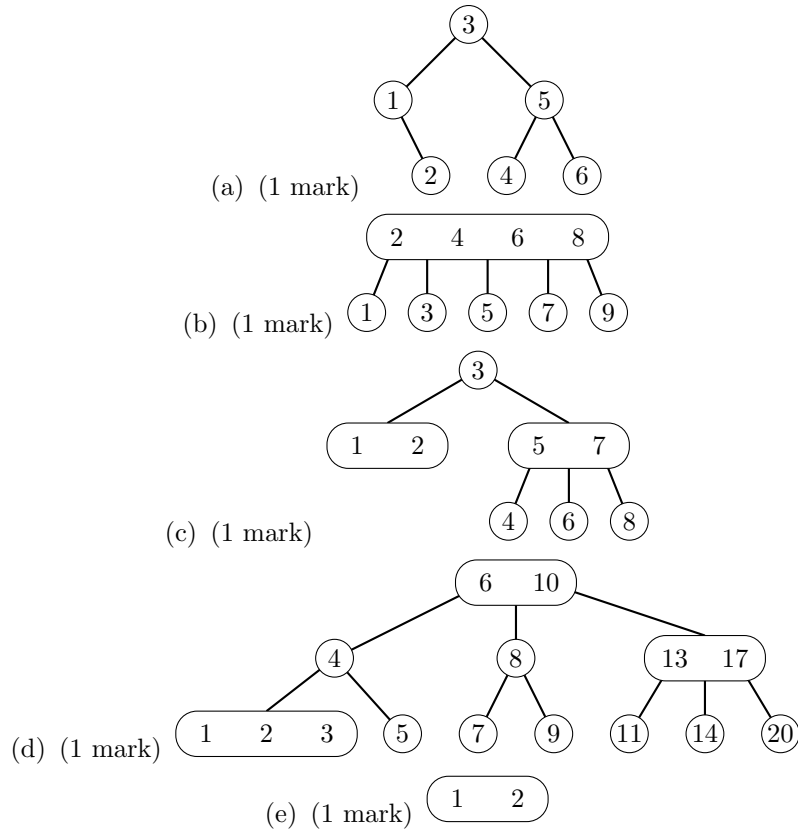
6. (5 marks) (Short question.) Consider the 11-vertex weighted graph G below.



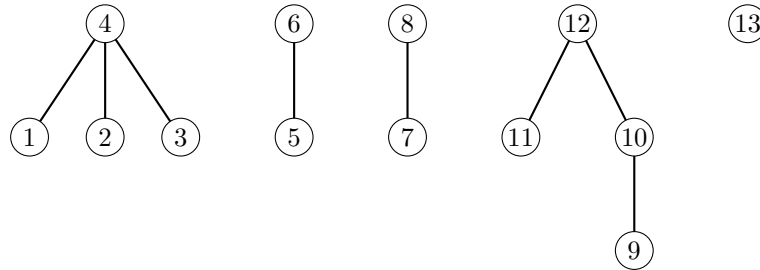
What is the weight of a minimum spanning tree of G ?

- A. 12.
- B. 13.
- C. 14.
- D. G doesn't have a minimum spanning tree.
- E. None of the above.

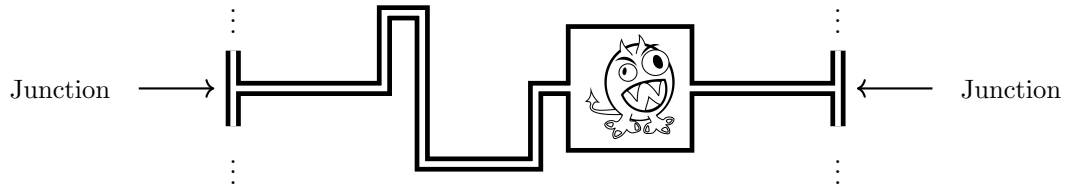
7. (5 marks) (Short question.) Which of the following are valid 2-3-4 trees?



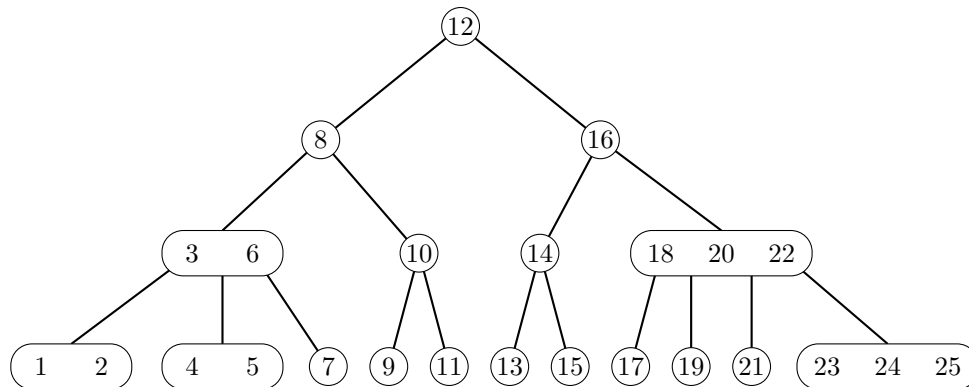
8. (5 marks) (Short question.) Consider the unoptimised version of the union-find data structure, in which a sequence of n operations takes $O(n \log n)$ time rather than $O(n\alpha(n))$ time. Suppose the current state of the data structure is as follows:



- (a) (1 mark) What is the result of **Find**(13)?
- (b) (1 mark) What is the result of **Find**(9)?
- (c) (2 marks) Suppose we now apply the operations **Union**(6,8), **Union**(1,7) and **Union**(10,4) to the data structure, in this order. How many components does the data structure now have?
- (d) (1 mark) Again after applying the **Union** operations above, what is the maximum depth of any component of the data structure? (The answer does not depend on how ties are broken.)
9. (5 marks) (Short question.) EvilCo Incorporated designs dastardly mazes for enterprising B-movie villains. In their marketing material, they say that their mazes are so complicated that if you want to get from one point to another without retracing your steps, there's only ever one way to do it. If one of their mazes has j junctions, how many corridors between junctions does it have, and why?
- (Note this does **not** include e.g. dead ends, or the entrance and exit — only corridors between two junctions count. Your answer should be short.)

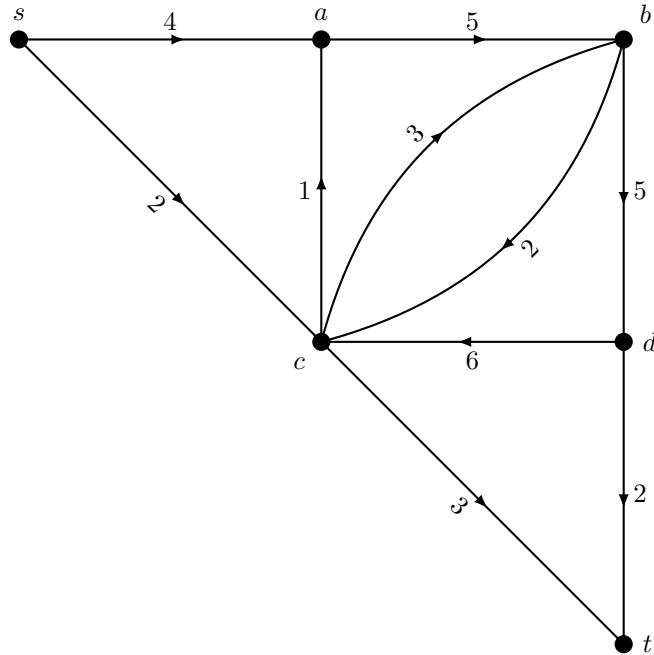


10. (5 marks) (Medium question.) You are the CEO of a sprawling corporation, which contains many overlapping product teams. You wish to set up a *representative meeting* — a meeting containing at least one representative from each product team — which is as small as possible, using the fact that a single person can act as a representative for all of the teams they're a member of. By reducing from vertex cover or otherwise, prove that it is NP-complete (under Karp reductions) to decide whether or not a k -person representative meeting exists, given a list of the teams and the value of k .
11. (5 marks) (Medium question.) Consider the 2-3-4 tree below:



After deleting the value 8 and inserting the value 30:

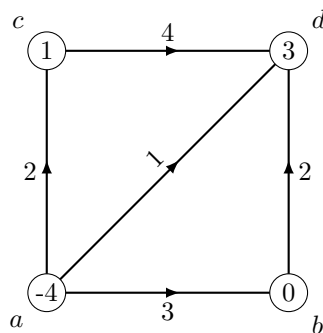
- (1 mark) What is the depth of the resulting tree?
 - (2 marks) How many 3-nodes does the resulting tree have?
 - (2 marks) While deleting 8 and inserting 30, how many fuse, transfer and split operations were performed in total?
12. (5 marks) (Short question.) Consider the following flow network (G, c, s, t) :



Let $A = \{s, b\}$ and $B = \{a, c, d, t\}$. What is the capacity $c^+(A)$ of the cut (A, B) ?

- A. 8.
- B. 13.
- C. 21.
- D. (A, B) is not a cut.
- E. None of the above.

13. (5 marks) (Short question.) Consider the following circulation network.



Which of the following is a valid circulation f ?

- A. $f(a, b) = 2$; $f(a, c) = 0$; $f(a, d) = 1$; $f(b, d) = 2$; $f(c, d) = 0$.
- B. $f(a, b) = 2$; $f(a, c) = 2$; $f(a, d) = 0$; $f(b, d) = 1$; $f(c, d) = 1$.
- C. $f(a, b) = 3$; $f(a, c) = 1$; $f(a, d) = 0$; $f(b, d) = 3$; $f(c, d) = 0$.
- D. $f(a, b) = 1$; $f(a, c) = 2$; $f(a, d) = 1$; $f(b, d) = 1$; $f(c, d) = 1$.
- E. None of the above, or more than one of the above.

14. (5 marks) (Short question.) Mark each of the following statements true or false.

- (a) (1 mark) The problem of outputting a size- k vertex cover in a graph G if one exists or **No cover** otherwise, given G and k as inputs, is NP-hard.
 - (b) (1 mark) The problem of outputting a size- k vertex cover in a graph G if one exists or **No cover** otherwise, given G and k as inputs, is in NP.
 - (c) (1 mark) Because SAT is an NP-complete problem, it is never practical to solve a 1,000-variable instance of SAT.
 - (d) (1 mark) If we had working quantum computers, we know how to use them to solve NP-complete problems in polynomial time.
 - (e) (1 mark) If a problem has instance size n and parameter k , then an algorithm with running time $O(n^{\sqrt{k}})$ shows that the problem is fixed-parameter tractable.
15. (5 marks) (Short question.) Consider an instance of weighted interval scheduling with interval set $\mathcal{R} = [(1, 5), (4, 6), (2, 11), (6, 13), (11, 16), (14, 19), (14, 21), (18, 21), (20, 25)]$ and weight function w given by

$$\begin{array}{llllll} w(1, 5) = 3, & w(4, 6) = 5, & w(2, 11) = 7, & w(6, 13) = 3, & w(11, 16) = 3, \\ w(14, 19) = 4, & w(14, 21) = 6, & w(18, 21) = 5, & w(20, 25) = 3. \end{array}$$

What is the maximum possible weight of a compatible set?

- A. 13.
 - B. 14.
 - C. 15.
 - D. 16.
 - E. None of the above.
16. (5 marks) (Short question.) You are captain of a spaceship in the far future, and you wish to traverse the galaxy from Caldari Prime to Molea Cemetary, hopping from planet to planet on the way while spending as few resources and making as much profit through trade on the way as possible. Your crew is troublesome — they require that each stop brings you strictly closer to your destination, or they will mutiny, and on arrival at a new planet they will go on shore leave and consume any goods left in your hold which you do not immediately sell. You are given a list of planets, their locations, and the (possibly negative) cost of travelling from one planet directly to another. Which of the following algorithms would be the best fit for this situation?
- A. Breadth-first search.
 - B. Dijkstra's algorithm.
 - C. The Bellman-Ford algorithm.
 - D. Depth-first search.
 - E. There is no unique best choice from the above options.
17. (10 marks) (Short question.) You have been placed in charge of overseeing plywood distribution in the USSR. You are given a list of production facilities F_1, \dots, F_n and destinations D_1, \dots, D_m . Each facility F_i can produce at most p_i units of plywood per day, and each destination D_j requires at least r_j units of plywood per day. The cost of moving one unit of plywood from facility F_i to destination D_j is $c_{i,j}$, and you may assume this scales linearly (so that e.g. the cost of moving half a unit is $c_{i,j}/2$). You wish to ensure that the requirements of each destination are met while spending as little on transportation as possible. Formulate this as a linear programming problem. (Your answer does not have to be in standard form, and you do not have to justify it.)
18. (10 marks) (Medium question.) You are given a directed graph G with vertex set $\{v_1, \dots, v_n\}$ which has the following property: for any edge $(v_i, v_j) \in E(G)$, we have $i < j$. That is, edges are always directed from vertices with lower indices to vertices with higher indices. You wish to find the length of

a **longest** path from v_1 to v_n , returning **Null** if no such path exists — recall that the length of a path is the number of **edges** it contains. Fill in the blanks of the following dynamic programming algorithm.

```

1 begin
2   Initialise  $A$  to an empty array indexed by  $1, \dots, n$ .
3   Set  $A[n] \leftarrow \text{BLANK}$ .
4   for  $i = n - 1$  to  $1$  do
5     Let  $S \leftarrow \{j : v_j \in N^+(v_i)\}$ .
6     if  $A[j] = \text{Null}$  for all  $j \in S$  then
7       Set  $A[j] \leftarrow \text{BLANK}$ .
8     else
9       Let  $X$  be the BLANK of the non-Null values of  $\{A[j] : j \in S\}$ .
10      Set  $A[i] \leftarrow X$  BLANK BLANK.
11  Return  $A[\text{BLANK}]$ .
```

Options for each blank: 0, 1, Null, +, −, maximum, minimum, none of the above.

19. (10 marks) (Medium question.) Given three distinct literals x , y and z , a *ONE clause* $\text{ONE}(x, y, z)$ evaluates to **True** if and only if **exactly one** of x , y and z evaluates to **True**. A *lonely formula* is a conjunction of ONE clauses, e.g.:

$$\text{ONE}(a, b, c) \wedge \text{ONE}(\neg a, b, \neg c) \wedge \text{ONE}(b, c, \neg d).$$

The decision problem LONELY-SAT asks whether a ONE formula given as part of the input is satisfiable. There is a valid Karp reduction from 3-SAT to LONELY-SAT which replaces each OR clause $(x \vee y \vee z)$ of a 3-SAT instance with a LONELY-SAT gadget containing four new variables a , b , c and d . Which of the following gadgets will work?

- A. $\text{ONE}(x, a, b) \wedge \text{ONE}(y, b, c) \wedge \text{ONE}(z, c, d)$.
- B. $\text{ONE}(\neg x, a, b) \wedge \text{ONE}(y, b, c) \wedge \text{ONE}(\neg z, c, d)$.
- C. $\text{ONE}(x, a, b) \wedge \text{ONE}(\neg y, b, c) \wedge \text{ONE}(z, c, d)$.
- D. $\text{ONE}(\neg x, a, \neg b) \wedge \text{ONE}(y, b, c) \wedge \text{ONE}(\neg z, \neg c, d)$.
- E. None of the above, or more than one of the above.

Section 2

20. (10 marks) (Medium question.) Let t be a natural number. Let G be an n -vertex graph with $2t$ vertices of odd degree and $n - 2t$ vertices of even degree. Prove, by induction on t or otherwise, that we can write $E(G)$ as a **disjoint** union of edge sets

$$E(G) = E(P_1) \cup \dots \cup E(P_t) \cup E(H),$$

where P_1, \dots, P_t are paths in G and H is a graph whose vertices all have even degree.

21. (10 marks) **Important:** The two parts of this question are completely independent — they have been put together for technical Blackboard reasons. Doing part a) will not help you with part b).
- (a) (5 marks) (Long question.) Suppose that a connected n -vertex, m -edge graph G (given in adjacency-list form) contains two vertices x and y which are distance $d(x, y) \geq n - c$ apart, for some integer $c \geq 1$. Give an algorithm which, given G , x , and an integer k as input, decides whether or not G contains a k -vertex clique in time $O(m + n + c^k)$. Briefly explain why it works and why it runs in the claimed time. (You do not need to give full pseudocode — an informal description is fine.)

Hint: You may find it helpful to use a BFS tree.

- (b) (5 marks) (Long question.) Let $n \geq 8$ be an even integer. Using Dirac's theorem or otherwise, prove that any n -vertex graph whose vertices all have degree at least $\frac{n}{2} + 3$ must contain a spanning subgraph whose vertices all have degree exactly 5. (Your answer shouldn't need to be more than a paragraph or two.)
22. (10 marks) **Important:** The two parts of this question are completely independent — they have been put together for technical Blackboard reasons. Doing part a) will not help you with part b).
- (a) (5 marks) (Long question.) You are teaching a class of schoolchildren C_1, \dots, C_n in the pre-COVID era. The end of term is approaching, and you have bought them a large tub of chocolates — you plan to give them four chocolates each, and eat the rest yourself. However, there are many types T_1, \dots, T_k of chocolate in the tub, and not every child likes every type of chocolate — a given child C_i is willing to accept only chocolates from a set $S_i \subseteq \{T_1, \dots, T_k\}$. The tub is also not bottomless — for each $i \in [k]$, there are only $t_i > 0$ chocolates of type T_i . Give an algorithm which, given T_1, \dots, T_k , t_1, \dots, t_k and S_1, \dots, S_n , assigns four chocolates to every child in polynomial time if possible and returns **Impossible** otherwise.
- (b) (5 marks) (Long question.) Your company has been tasked with producing a set of widgets w_1, \dots, w_t , using a 3D printer to make the parts. Each widget w_i must be first printed in time p_i , then assembled in time a_i . You have enough workers to be able to assemble as many widgets as you like in parallel, but you can only print one widget at a time. You would like to find an order of printing the widgets which allows you to finish assembling them all as quickly as possible. Give a greedy algorithm which, given p_1, \dots, p_t and a_1, \dots, a_t , will output an optimal widget order in $O(t \log t)$ time. Give a careful proof that it works (e.g. using a greedy-stays-ahead or exchange argument).
23. (10 marks) (Medium question.) Consider the following greedy algorithm for SAT.

Input : A CNF formula F with variables x_1, \dots, x_n and clauses C_1, \dots, C_m .
Output: A satisfying assignment if F is satisfiable, No otherwise.

```

1 begin
2   for  $i = 1$  to  $n$  do
3     Let  $a_i$  be the number of clauses  $C_j$  containing  $x_i$ .
4     Let  $b_i$  be the number of clauses  $C_j$  containing  $x_i$  such that every other literal in  $C_j$  has
       already been set to False.
5     Let  $c_i$  be the number of clauses  $C_j$  containing  $\neg x_i$ .
6     Let  $d_i$  be the number of clauses  $C_j$  containing  $\neg x_i$  such that every other literal in  $C_j$  has
       already been set to False.
7     if  $b_i > 0$  and  $d_i > 0$  then
8       | Return No.
9     else if  $b_i > 0$  then
10      | Set  $x_i$  to True.
11     else if  $d_i > 0$  then
12      | Set  $x_i$  to False.
13     else if  $a_i \geq c_i$  then
14      | Set  $x_i$  to True.
15     else
16      | Set  $x_i$  to False.
17   Return the values of  $x_1, \dots, x_n$ .
```

Give a counterexample to show that this algorithm does not work, and briefly explain it.