CS31005 Algorithms–II

Mid-Semester Test

Evaluation Notes

2. 8 marks for flow modeling + 2 marks for condition of feasibility and showing allocation. Variations other than the one shown in the solution given are possible, and have been given due credit if correct. Many of you have used the $E[i]$ and $S[j]$ values as capacities on the flow graph. If so, you lost 50% immediately as then you are just trying to distribute total students among total capacity, which can make a class break across different rooms and at different times, which implies that a room may have more than one class scheduled in it in a single timeslot. So for such answers, you are graded out of 5 marks, and then any other deduction made from that if additional things are wrong.

3. Most solutions from students start with the (hypothetical) premise that there are two men (say, m_1 and m_2) who are paired with their respective last choices (say, w_1 and w_2), and try to arrive at a contradiction. While this is a promising start, a contradiction is not really needed for proving the assertion. More importantly, the contradictions that these solutions arrived at are not really so.

By the choice of w_1 and w_2, the preference lists of m_1 and m_2 are as follows.

$m_1 : \ldots w_2, \ldots, w_1$
$m_2 : \ldots w_1, \ldots, w_2$

Up to this point, the arguments are fine. But then, what is wrong about the following situation?

m_1 proposes to w_2, and is accepted for the time being.
m_2 proposed to w_1 who is already engaged to some m_3 better than m_2.
So w_1 rejects m_2 leaving m_2 unengaged.
m_2, after going through other rejections/replacements (if any), eventually proposes to w_2.
Suppose that w_2 is still engaged to m_1, but m_2 is a better choice for her.
So w_2 replaces m_2 by m_1, so m_1 becomes unengaged again.
m_1, after going through other rejections/replacements (if any), eventually proposes to w_1.
Suppose that w_1 is still engaged to m_3, but m_1 is a better choice for her.
So w_1 accepts m_1.

In the process, m_3 gets unengaged again, but that does not matter, because two men have already proposed to their respective last choices. Moreover, what we learn from the above transcript about the choices of w_1 and w_2 is the following.

$w_1 : \ldots, m_1, \ldots, m_3, \ldots, m_2, \ldots$
$w_2 : \ldots, m_2, \ldots, m_1, \ldots$

There is no contradiction in these choice lists.

The solutions fail to clearly resolve such a possibility. Indeed, if w_1 and w_2 are not the last choices of m_1 and m_2, such a possibility is very real.

The correct observation is that when a man proposes to his last choice, the algorithm terminates. If that does not happen, then the situation in the above transcript can happen too. So a contradiction is not necessary for the proof, and, more importantly, not sufficient too for the solutions of the students.

In the solutions, several incorrect claims are made. First, a man proposing to his last choice is the last choice of some/every woman. As a counterexample, take $n \geqslant 3$, and consider the following preference lists.

m_1	w_1			$\cdots$	
m_2	w_2			$\cdots$	
m_3	w_3			$\cdots$	
$\vdots$					
m_{n-1}	w_{n-1}			$\cdots$	
m_n	w_1	w_2	w_3	$\cdots$	w_n

w_1	m_1	m_n	$\cdots$
w_2	m_2	m_n	$\cdots$
w_3	m_3	m_n	$\cdots$
$\vdots$			
w_{n-1}	m_{n-1}	m_n	$\cdots$
w_n	m_n		$\cdots$

Second, if multiple men are paired with their respective last choices, then the matching is not stable. The man-optimal version of the GS algorithm may give a matching in which every woman gets her last choice (it is straightforward to illustrate this by an example). Therefore the woman-optimal version (women propose, men take decisions) of the algorithm can give stable matchings in which every man gets his last choice.

A few solutions swapped the pairs (m_1, w_1) and (m_2, w_2) by (m_1, w_2) and (m_2, w_1), and use the man-optimality of the GS algorithm. While this swapping gives a man-better pairing, a careful argument remains to establish that the new matching is stable too. An unstable pair is a pair (m, w) *not* in the matching, so looking only at the local rearrangement does not prove the stability in the swapped matching. In other words, you need to show that no (m, w) pair not in the new matching is unstable.

Penalties in the range 4–7 are applied on these errors depending on subjective judgments of the severity of the errors.

4. A broad distribution of the 12 marks among the various components is as follows.

(a) Identification of the events: 2 marks
(b) Handling of the events: 4 marks
(c) Data structure for event queue, with justification: 2 marks
(d) Data structure for sweep-line information, with justification: 2 marks
(e) Justification for the $O((m+n)O(m+n))$ running time: 2 marks

However, the parts are so intertwined that it is difficult to follow this breaking of marks rigorously. For example, you should mention somewhere how a house is searched in the "list" of active regions. This may be anywhere in (b), (d), or (e). If it reads like a naive search (which some solutions did by looking at all houses as soon as a left-end of a square is encountered), a significant penalty is imposed. Moreover, if the events are not handled, parts (c), (d), and (e) make no sense. What is the running time or a data structure if there is no event-handling algorithm in (a) and/or (b)?

So the evaluation is based on overall correctness using the above distribution of marks as a basic guideline.

Although a balanced binary search tree meets the timing requirements for implementing the event queue, a sorted array or a min-heap is practically much more efficient than potentially huge height-balancing overheads of a BBST. Likewise, the active regions are non-overlapping, so there is no need to use interval trees to implement the sweep-line-information data structure. Use a data structure that is just sufficient for the purpose. There is no need to store both the upper and the lower y-coordinates of an active region, but many solutions rely upon that, and no penalty is imposed for this.

5. Show that COMPLEMENTSAT is in NP: 2 marks. You need to mention clearly what a certificate for an accept instance of COMPLEMENTSAT is (1 mark), and how it is verified (1 mark). Alternatively, you can design a non-deterministic poly-time algorithm.

Reduction from a known NP-complete problem Q to COMPLEMENTSAT: 8 marks. Clearly show how an instance I of Q is converted to an instance ϕ for COMPLEMENTSAT (2 marks). You also need to supply a two-way proof to establish that $I \in \text{Accept}(Q)$ if and only if $\phi \in \text{Accept}(\text{COMPLEMENTSAT})$ (6 marks). If the construction is correct but only a one-way proof is given, then this is evaluated in 3 marks. If the construction does not work, then no credit is given in the proof part (even if one direction of the proof is true). Moreover, if there is no clear construction, the proof makes no sense and deserves no credit.

Natural choices for Q are SAT or CNFSAT. An unconventional choice is NAESAT, where the conversion of I to ϕ is just taking $\phi = I$. However, note that the two problems NAESAT and COMPLEMENTSAT are not the same. NAESAT takes as input a CNF formula, whereas COMPLEMENTSAT takes as input any Boolean formula.