Algorithmics II (H) Exam April / May 2023 – Solutions

1. **(a)** **(i)** The following operations would take place involving the temporary convex hull, assuming that p_7 is chosen as the pivot:

1. add p_7
2. add p_{12}
3. add p_8
4. remove p_8
5. add p_3
6. add p_9
7. add p_4
8. remove p_4
9. remove p_9
10. add p_{11}
11. add p_1
12. remove p_1
13. add p_5
14. add p_6
15. remove p_6
16. add p_{10}
17. remove p_{10}
18. add p_2

(ii) The convex hull comprises p_7, p_{12}, p_3, p_{11}, p_5, p_2.

(b) **(i)** Let t be the predecessor of q on the temporary convex hull. We must make a "left turn" at q when travelling from t to q to r, since t was discovered before q during the anticlockwise scan phase and r has a larger x-coordinate than q.

(ii) Let s be the second point added to the temporary convex hull (after r). We must make a "left turn" at r when travelling from q to r to s, since s was discovered before q during the anticlockwise scan phase and r has a larger x-coordinate than each of q and s.

2. **(a)** The recursive method `recExtent` can be completed as follows:

```
recExtent(int i, int j, int I, int J)
{  if (S(i,j)==0)
      output "X: ["+(i+1)+","+I+"]  Y: ["+(j+1)+","+J+"]";
   else if (X(i)==Y(j))
      recExtent(i-1,j-1,I,J);
   else
   {  if (S(i,j)==S(i-1,j-1)-1)
         recExtent(i-1,j-1,I,J);
      if (S(i,j)==S(i-1,j)-1)
         recExtent(i-1,j,I,J);
      if (S(i,j)==S(i,j-1)-1)
         recExtent(i,j-1,I,J);
   }
}
```

(b) An optimal alignment of (the unique) highest scoring local similarity of X and Y is as follows:

```
G C A T A T C G
| | | | | | | |
m m m s m d m m
| | | | | | | |
G C A G A   C G
```

3. **(a)** The remaining preference lists upon termination of the algorithm are as follows:

Students' preferences	Lecturers' preferences
s_1: ℓ_4 ℓ_1 ℓ_2 ℓ_3	ℓ_1: s_3 s_4 s_1 s_2
s_2: ℓ_1	ℓ_2: s_4 s_1 s_3
s_3: ℓ_2 ℓ_1	ℓ_3 s_1 s_4
s_4: ℓ_3 ℓ_1 ℓ_2	ℓ_4: s_1

(b) Clearly M is a matching, since upon termination, each student is assigned exactly one lecturer and each lecturer is assigned at most one student. Suppose that s prefers ℓ to their partner in M. Then ℓ was deleted from s's list during the algorithm's execution, so s was deleted from ℓ's list. That happened because some student s' became matched to ℓ, where ℓ prefers s' to s. Hence ℓ's partner in M must be s' or better, and hence better than s. Thus (s, ℓ) does not block M, and since (s, ℓ) was arbitrary, M must be stable.

4. **(a)** Given an instance I of PARTITION comprising integers $x_1, x_2, \ldots, x_n$, create the following instance I' of JSSP. Let $J=\{j_1, j_2, \ldots, j_n\}$ and create n identical machines. Let $t_i = x_i$ for all i $(1 \leq i \leq n)$ and let $T=½\sum_{i\in N} x_i$.

Now suppose that A is an approximation algorithm for JSSP with a performance guarantee $<3/2$.

Suppose firstly that I admits a partition. Then $m^*(I') = 2$, so A gives an assignment of jobs to machines using $< 3/2\ m^*(I')$ machines, i.e., using 2 machines.

Conversely suppose that I does not admit a partition. Then $m^*(I') \geq 3$. Hence A gives an assignment of jobs to machines using at least 3 machines.

Thus the existence of A gives a polynomial-time algorithm for PARTITION, which is a contradiction unless P=NP.

(b) Process the jobs in some order. For each job j, if some machine m already in use has enough spare time to accommodate j, assign j to m, otherwise assign j to a new machine.

(c) JSSP does have an approximation algorithm with performance guarantee 3/2. This is because JSSP is equivalent to the Minimum Bin Packing problem (which is approximable within 3/2 using the First Fit Decreasing algorithm, though this reasoning was not required as part of the answer).