Xi'an Jiaotong-Liverpool University 西交利物浦大學

PAPER CODE	EXAMINER	DEPARTMENT	TEL
INT102	Wenjin Lu	Intelligent Science	1505

2nd SEMESTER 2021/22 EXAMINATIONS (RESIT Open Book)

BACHELOR DEGREE – Year 2

ALGORITHMIC FOUNDATIONS AND PROBLEM SOLVING

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

READ THE FOLLOWING CAREFULLY:

1. The paper consists of Part A and Part B. Answer all questions in both parts. Total marks available are 100. Marks for this examination account for 80% of the total credit for INT102.

2. In Part A, each of the questions comprises 5 statements, for which you should select the one most appropriate answer.

3. Answers to questions in Part B should be written in the answer script.

4. This is an OPEN BOOK examination. You can reference textbooks and notes but discuss with other students in any way is not allowed.

5. The time of the exam is strictly limited to 2 hours.

6. For students who take the exam online, at the end of the examination, be absolutely sure to submit your answer via Learning Mall. The time for submission of your answer via Learning Mall is strictly limited to 15 minutes. Once the time is over, the submission link will be closed.

7. All answers must be in English.

PART B (40 marks)

Question 1 (13 marks)

1. Suppose T(n) denotes the worst case time complexity of the merge-sort algorithm on n numbers.

   a) Explain why T(n) can be described by the following recurrence. **7**

$$T(n) = \begin{cases} 1 & if \quad n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & if \quad n > 1 \end{cases}$$

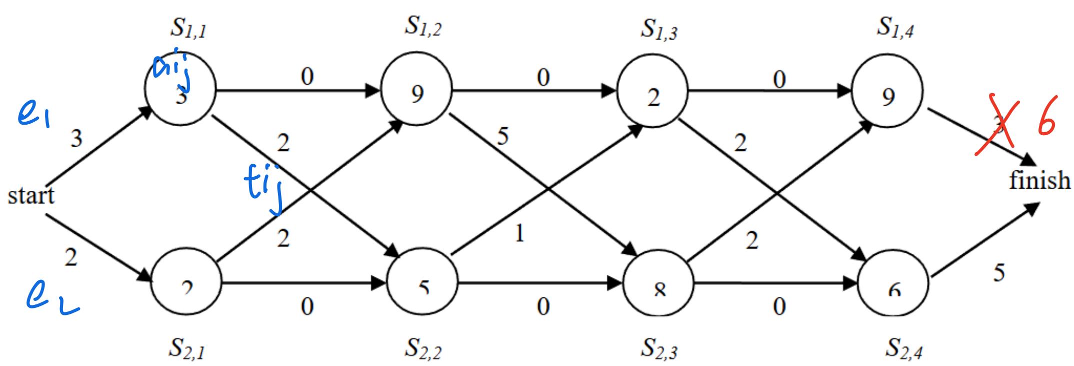
   b) Show that T(n) = O(nlog n) by the substitution method. (Hint: show that T(n) ≤ 2n log n for n≥ 2 by Mathematical Induction.) **6**

Question 2 (27 marks)

Suppose there are two assembly lines each with n stations, for each j, $1 \le j \le n$, let $S_{i,j}$ denote the jth station in line i (i = 1, 2). For each i (i = 1, 2), let

- e_i be the entry time into line i.
- x_i be the exit time from line i.
- $a_{i,j}$ be the assembly time at $S_{i,j}$
- $t_{1,j}$ be the transfer time from station $S_{1,j}$ to station $S_{2,j+1}$ and $t_{2,j}$ be the transfer time from station $S_{2,j}$ to station $S_{1,j+1}$.

The following is an example of two assembly lines, each line has 4 stations with e_1=3, e_2=2, x_1=6, x_2=5. The assembly time is given in the circle representing the station and the transfer time is given next to the edge from one station to another.



1. Let $f_1[j]$ and $f_2[j]$ be the fastest time to get through stations $S_{1,j}$ and $S_{2,j}$, respectively. Using dynamic programming technique, derive a recursive definitions of $f_1[j]$ and $f_2[j]$ and the fastest time f* needed to get through the assembly line. **6**

$f_{11} = e_1 + a_{11} = 6$

$f_{21} = e_2 + a_{21} = 4$

$f_{1j} = \min(a_{1j} + t_{2j-1} + f_{2j-1},\ a_{1j} + f_{1j-1})$

$f_{2j} = \min(a_{2j} + t_{1j-1} + f_{1j-1},\ a_{2j} + f_{2j-1})$

$f^* = min(f_1[4] + x_1, f_2[4] + x_2)$

2. For the assembly lines give above, fill in the table of the fastest time $f_i[j]$needed to get through station $S_{i,j}$. Show all the intermediate steps in computing these values. **6**

j	$f_1[j]$	$f_2[j]$
1	… 6	… 4
2	… 15	… 9
3	… 12	… 17
4	… 21	… 20

3. For the assembly lines give above, what is the fastest time f* needed to get through the assembly line? **2**

25

4. For the assembly lines give above, which stations should be chosen? **6**

$S_{21} \to S_{12} \to S_{13} \to S_{14}$

5. Based on the recurrence relation obtained in a), write a pseudo code of the bottom-up dynamic programming algorithm with running time O(n) for the n stations assembly line scheduling problem. **7**

END OF THE PAPER

```
// 初始化第一个站点的最优值和来源
f1[1] ← e1 + a1[1]
f2[1] ← e2 + a2[1]
line1[1] ← 1
line2[1] ← 2

// 自底向上计算：对每个 j=2…n，决定留在本线还是切换过来
for j from 2 to n do
  // 计算到 S1,j 的最短时间
  if f1[j-1] + a1[j] ≤ f2[j-1] + t2[j-1] + a1[j] then
     f1[j] ← f1[j-1] + a1[j]
     line1[j] ← 1       // 最优前驱来自同线
   else
     f1[j] ← f2[j-1] + t2[j-1] + a1[j]
     line1[j] ← 2       // 最优前驱来自另一线
   end if

   // 计算到 S2,j 的最短时间（同理）
   if f2[j-1] + a2[j] ≤ f1[j-1] + t1[j-1] + a2[j] then
     f2[j] ← f2[j-1] + a2[j]
     line2[j] ← 2
   else
     f2[j] ← f1[j-1] + t1[j-1] + a2[j]
     line2[j] ← 1
   end if
end for

// 考虑出口时间，选最终哪条线最优
if f1[n] + x1 ≤ f2[n] + x2 then
   f* ← f1[n] + x1
   lastLine ← 1
else
   f* ← f2[n] + x2
   lastLine ← 2
end if

// 回溯：从终点往前重建各站的选线结果
line[n] ← lastLine
for j from n down to 2 do
   if line[j] = 1 then
     line[j-1] ← line1[j]
   else
     line[j-1] ← line2[j]
   end if
end for

return (f*, line[1..n])
```

Q1.

Answer:

1. **Divide.** Split an n-element array into two subarrays of size $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$.
2. **Recurse.** Recursively sort each half. Each half costs $T(\lfloor n/2 \rfloor)$, so together $2\,T(\lfloor n/2 \rfloor)$.
3. **Merge.** Merge the two sorted halves by repeatedly comparing their front elements and moving the smaller one into the output array. This takes $\Theta(n)$ time.
4. **Base case.** If $n = 1$, the array is already sorted in constant time (we write $T(1) = 1$).

Summing these costs gives the stated recurrence.

(b) Show by induction that $T(n) = O(n \log n)$.

We prove by strong induction that
$T(n) \le 2\,n \log n \quad \text{for all } n \ge 2.$

1. **Base case** $(n = 2)$:

$T(2) = 2\,T(1) + 2 = 2 \cdot 1 + 2 = 4, \quad 2 \cdot 2 \log 2 = 4 \cdot 1 = 4, \quad \text{so } T(2) \le 2 \cdot 2 \log 2.$

2. **Inductive hypothesis.** Assume $T(k) \le 2\,k \log k$ for all $2 \le k < n$.
3. **Inductive step.** For $n > 2$,

$$\begin{aligned} T(n) &= 2\,T(\lfloor n/2 \rfloor) + n \\ &\le 2\Big[2\lfloor \tfrac{n}{2} \rfloor \log\lfloor \tfrac{n}{2} \rfloor\Big] + n && \text{(by IH, since } \lfloor n/2 \rfloor < n) \\ &\le 4 \cdot \tfrac{n}{2}(\log(n/2)) + n && \big(\lfloor n/2 \rfloor \le n/2,\ \log\lfloor n/2 \rfloor \le \log(n/2)\big) \\ &= 2\,n(\log n - 1) + n = 2\,n\log n - n \le 2\,n \log n. \end{aligned}$$

4. **Conclusion.** By induction, $T(n) \le 2\,n \log n$ for all $n \ge 2$. Hence $T(n) = O(n \log n)$.