

---

UNIVERSITI SAINS MALAYSIA

Second Semester Examination  
2014/2015 Academic Session

June 2015

**CCS592 – Advanced Algorithms and Complexity**  
*[Algoritma Lanjutan & Kekompleksan]*

Duration : 2 hours  
*[Masa : 2 jam]*

---

**INSTRUCTIONS TO CANDIDATE:**

***[ARAHAN KEPADA CALON:]***

- Please ensure that this examination paper contains **FOUR** questions in **EIGHT** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **LAPAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

*[Jawab **SEMUA** soalan.]*

- You may answer the questions either in English or in Bahasa Malaysia.

*[Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]*

- In the event of any discrepancies, the English version shall be used.

*[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]*

1. (a) You have been assigned into a team to build the next generation Walkman (music player) by Soni. You are asked to implement the processes described below. When powered up, the Walkman will be initialized to an initial state. When a user inserts a tape, music will be played. When the user presses on the pause button when the music is playing, the Walkman will stop playing music. But if the user presses eject while music is playing, the Walkman will eject the tape and go back to the initial state. When the Walkman is paused, user can press the pause button to continue playing music or eject button to eject the tape.

*Anda telah ditugaskan ke dalam satu pasukan untuk membina Walkman (pemain music) generasi baru oleh Soni. Anda diminta untuk melaksanakan proses yang dinyatakan berikut. Apabila dihidupkan, Walkman akan diawalkan kepada suatu keadaan permulaan. Apabila pengguna memasukkan pita, muzik akan dimainkan. Apabila pengguna menekan butang jeda ketika muzik dimainkan, Walkman akan berhenti memainkan muzik. Tetapi jika pengguna menekan butang memacu semasa muzik sedang bermain, Walkman akan mengeluarkan pita dan kembali kepada keadaan permulaan. Ketika Walkman dijedakan, pengguna boleh tekan butang jeda untuk meneruskan muzik atau menekan butang memacu untuk mengeluarkan pita.*

Implement the rules explained above for the Walkman using appropriate programming design pattern with Java or other appropriate object oriented programming language.

*Laksanakan peraturan yang dijelaskan di atas untuk Walkman menggunakan corak reka bentuk pemrograman yang sesuai dengan Java atau bahasa pengaturcaraan berorientasikan objek lain yang sesuai.*

(40/100)

- (b) The **travelling salesman problem (TSP)** asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

*Masalah jurujual kembara (TSP) bertanya soalan berikut: Diberi senarai bandar-bandar dan jarak antara setiap pasang bandar, apakah laluan terpendek untuk melawat setiap bandar sekali dan kembali semula ke bandar permulaan?*

- (i) Describe how Graham scan works.

*Jelaskan bagaimana pengimbas Graham dilaksanakan.*

- (ii) Analyze and explain how Graham scan can be used to find a reasonably good solution for the travelling salesman problem.

*Analisis dan jelaskan bagaimana pengimbas Graham boleh digunakan untuk mendapatkan penyelesaian yang munasabah untuk masalah jurujual kembara.*

(60/100)

2. The Chinese postman problem is also known as the route inspection problem. In this problem, there is a mailman who needs to deliver mail to a certain neighbourhood. The mailman is unwilling to walk far, so he wants to find the shortest route through the neighbourhood that meets the following criteria:

*Masalah posmen Cina juga dikenali sebagai masalah pemeriksaan laluan. Dalam masalah ini, terdapat seorang posmen yang perlu menghantar surat ke kawasan sekitar. Posmen tidak bersedia untuk berjalan jauh, jadi dia ingin mencari laluan terpendek untuk melalui kawasan sekitar yang memenuhi kriteria berikut:*

- The route is a closed circuit (it ends at the same point it starts).

*Laluan membentuk suatu litar tertutup (ia berakhir pada titik yang sama ia bermula).*

- Postman needs to go through every street at least once.

*Posmen perlu melalui setiap jalan sekurang-kurangnya sekali.*

- Chinese postman problem is a NP-Complete problem.

*Masalah posmen Cina adalah masalah NP-lengkap.*

- (a) Analyze the problem given and explain using an example what is the difficulty in solving this problem.

*Kaji masalah yang diberikan dan terangkan menggunakan contoh apa kesukaran dalam menyelesaikan masalah ini.*

(30/100)

- (b) What is a greedy algorithm?

*Apakah yang dimaksudkan dengan algoritma tamak?*

(20/100)

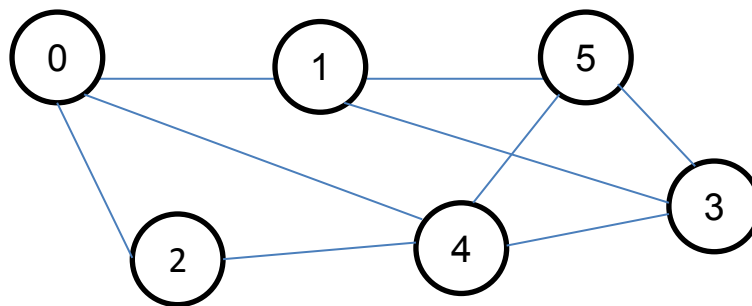
- (c) Propose a greedy algorithm that could give a reasonably optimal solution. Explain how the algorithm solve the given problem, then use an appropriate example to simulate the algorithm.

*Cadangkan algoritma tamak yang boleh memberikan penyelesaian yang cukup optimum. Terangkan bagaimana algoritma tersebut menyelesaikan masalah yang diberi, kemudian gunakan contoh yang sesuai untuk mensimulasikan algoritma yang dicadangkan.*

(50/100)

3. (a) Based on the graph below, justify whether the graph has the following properties or otherwise.

*Berdasarkan graf di bawah, jelaskan sama ada graf berkenaan mempunyai sifat-sifat berikut atau sebaliknya.*



- (i) A connected graph.  
*Sebuah graf terkait.*
- (ii) A complete graph.  
*Sebuah graf lengkap.*
- (iii) A bipartite graph.  
*Sebuah graf dwipihak.*
- (iv) A planar graph.  
*Sebuah graf sesatah.*
- (v) There is no articulation points.  
*Tidak terdapat titik persendian.*

(30/100)

- (b) For the graph in Question 3(a), the following information ( $v = \text{node}$ ,  $\text{DFSTreeParent}[v] = \text{the parent of } v \text{ in the depth-first search tree}$ ) is given:

*Bagi graf dalam Soalan 3(a), maklumat ( $v = \text{nod}$ ,  $\text{DFSTreeParent}[v] = \text{bapa } v \text{ dalam pepohon gelintaran kedalaman dahulu}$ ) berikut diberikan:*

$v$		0	1	2	3	4	5
$\text{DFSTreeParent}[v]$		2	0	4	5	-1	1

Identify the root of the depth-first search tree and sketch the tree obtained from the traversal.

*Kenal pasti akar pepohon penyusuran gelintaran kedalaman dahulu dan lakar pepohon yang diperoleh daripada penyusuran berkenaan.*

(20/100)

- (c) Given below is the pseudo-code for finding a shortest path based on the Dijkstra's algorithm.

*Diberi di bawah pseudokod untuk mencari sebuah lintasan terpendek berdasarkan algoritma Dijkstra.*

```

procedure Dijkstra( $G, a, w, T$ )
   $\text{Dist}[0:n-1]$  is an array initialised to  $\infty$ 
   $T$  is initialised to be tree consisting of the single vertex  $r$ 
   $\text{Dist}[r] \leftarrow 0$ 
  while  $\text{Cut}(T) \neq \emptyset$  do
    select an edge  $uv \in \text{Cut}(T)$  such that  $\text{Dist}[u] + w(uv)$  is minimised
    add vertex  $v$  and edge  $uv$  to  $T$ 
  endwhile
end Dijkstra

```

- (i) Why is this algorithm considered to be a greedy method?

*Mengapakah algoritma ini dianggap sebagai kaedah tamak?*

- (ii) How many times would the *while* loop be executed for a connected graph using this algorithm?

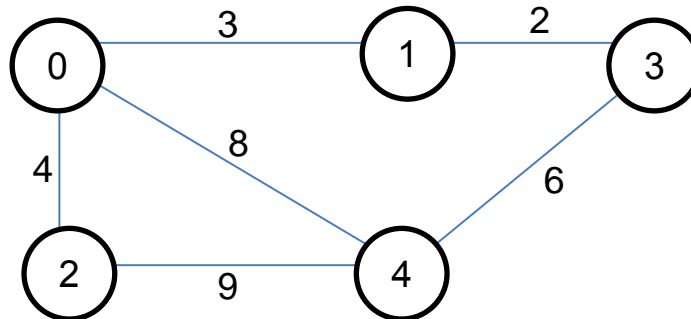
*Berapa kalikah gelung while dilakukan untuk graf terkait menggunakan algoritma ini?*

- (iii) The complexity of this algorithm depends on specific implementation details of the algorithm. Why?

*Kekompleksan algoritma ini bergantung kepada perincian pelaksanaan yang tertentu algoritma berkenaan. Mengapa?*

- (iv) Trace step by step how the algorithm works on the following graph.

*Surih langkah demi langkah bagaimana algoritma berkenaan dilaksanakan ke atas graf berikut.*



(50/100)

4. (a) Given the following pseudo-code for the naïve algorithm for string matching:

*Diberi pseudokod berikut untuk algoritma naïf untuk pemadanan rentetan:*

```

function NaiveStringMatcher(P[0:m-1], T[0:n-1])
  P[0:m-1] // a pattern string of length m (rentetan pola panjang m)
  T[0:n-1] // a text string of length n (rentetan teks panjang n)
  for s ← 0 to n-m do
    if T[s: s+m - 1] = P then
      return (s)
    endif
  endfor
  return (-1)
end NaiveStringMatcher
  
```

- (i) What will be the possible outputs of this pseudo-code and what exactly is the output for each case?

*Apakah output-output yang mungkin bagi pseudokod ini dan apakah output yang sebenar bagi setiap kes?*

- (ii) Trace the algorithm by giving a proper pattern that will show a worst-case situation when the text is "aaaaaaaaab".

*Surih algoritma berkenaan dengan memberi pola yang betul yang akan menunjukkan situasi kes terburuk apabila teks ialah "aaaaaaaaab".*

- (iii) Show that worst-case complexity of the algorithm is  $O(mn)$ .

*Tunjuk bahawa kekompleksan algoritma berkenaan ialah  $O(mn)$ .*

- (iv) How do Knuth-Morris-Pratt and Boyer-Moore algorithms improve the complexity of the naïve algorithm?

*Bagaimanakah algoritma Knuth-Morris-Pratt dan algoritma Boyer-Moore menambah baik kekompleksan algoritma naïf?*

(50/100)

- (b) Given below is the pseudo-code for the algorithm FFTRec for evaluating  $P(1)$ ,  $P(\omega)$ ,....., $P(\omega^{n-1})$ :

*Diberi di bawah ialah pseudokod untuk algoritma FFTRec untuk menilai  $P(1)$ ,  $P(\omega)$ ,....., $P(\omega^{n-1})$ :*

```

procedure FFTRec (a[0:n-1],n,  $\omega$ , b[0:n-1]) recursive
  if n =1 then
    b[0]  $\leftarrow$  a[0]
  else
    for j  $\leftarrow$  0 to n/2 -1 do
      Even[j]  $\leftarrow$  a[2*j]
      Odd[j]  $\leftarrow$  a[2*j+1]
    endfor
    FFTRec(Even[0:n/2-1], n/2,  $\omega^2$ , e[0:n/2-1])
    FFTRec(Odd[0:n/2-1], n/2,  $\omega^2$ , d[0:n/2-1])
    for j  $\leftarrow$  0 to n/2 -1 do
      b[j]  $\leftarrow$  e[j] +  $\omega^j$  * d[j]
      b[j + n/2]  $\leftarrow$  e[j] +  $\omega^j$  * d[j]
    endfor
  endif
end FFTRec

```

- (i) Define all the parameters of the above procedure.

*Takrifkan semua parameter bagi tatacara di atas.*

- (ii) Which part of the above procedure is the part where divide-and-conquer strategy is applied?

*Bahagian manakah dalam tatacara di atas merupakan tempat strategi bahagi dan tawan diterapkan?*

- (iii) Without going into mathematical detail of the above algorithm, describe how the divide-and-conquer strategy is used to reduce its complexity.

*Tanpa perincian matematik bagi algoritma di atas, huraikan bagaimana strategi bahagi dan tawan digunakan untuk mengurangkan kekompleksannya.*

(30/100)

- (c) Give with justification **two (2)** examples of graph algorithm that belong to the class of NP-Complete problems and **two (2)** examples of graph algorithm that do not belong to this class.

*Beri dengan penjelasan **dua (2)** contoh algoritma graf yang tergolong dalam kelas masalah NP-Lengkap dan **dua (2)** contoh algoritma graf yang tidak tergolong dalam kelas berkenaan.*

(20/100)