UNIVERSITI SAINS MALAYSIA

Peperiksaan Kursus Semasa Cuti Panjang
Sidang Akademik 2004/2005

Mei 2005

CPT201 – Reka Bentuk & Analisis Algoritma

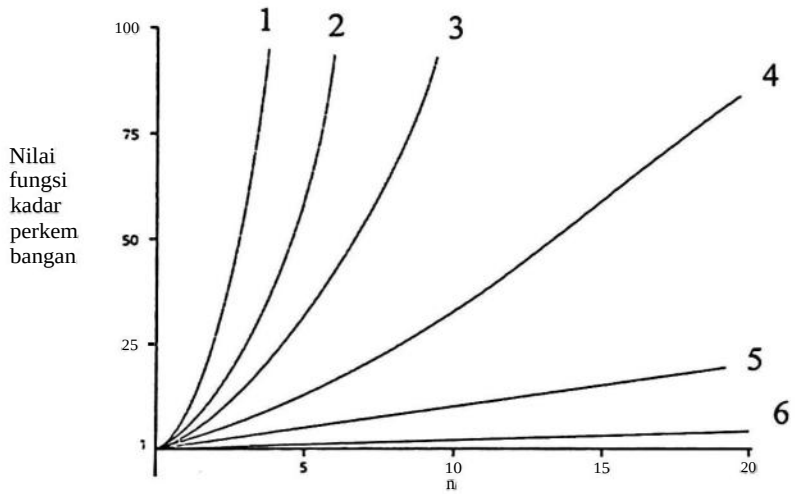
Masa : 2 jam

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi LIMA soalan di dalam ENAM muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
- Jawab mana-mana EMPAT (4) soalan.

---

1. (a) Labelkan lengkung 1 hingga 6 di bawah dengan fungsi perkembangan yang bersesuaian:



(20/100)

(b) Mengapakah kita perlu berhati-hati dalam kes-kes berikut?

(i) Kita tidak seharusnya menganalisis masa perlaksanaan sesuatu penyelesaian berdasarkan pelaksanaan tertentu.

(ii) Apabila sesuatu masalah itu kecil, janganlah kita membuang banyak masa menganalisis masalah berkenaan.

(iii) Apabila membandingkan kecekapan beberapa penyelesaian, pertimbangkan perbezaan yang signifikan sahaja.

(20/100)

(c) Berikut diberikan algoritma **quicksort** (isihan cepat) dan **mergesort** (isihan canturn):

```
quicksort (inout theArray:ItemArray, in first:integer,
                  in last:integer)
if (first < last)
{  partition (theArray) //Pemetakan - Sediakan tatasusunan
                        // theArray untuk panggilan rekursi
   quicksort(Si region of theArray) // Rantau Sx theArray
   quicksort(S2 region of theArray) // Rantau S2 theArray
}

mergesort(inout theArray:ItemArray, in first:integer,
                  in last:integer)
if (first < last)
{  mergesort(Left half of theArray) // Setengah kiri theArray
   mergesort(Right half of theArray)// Setengah kanan theArray
   merge(TheArray) //Pencantuman - Bersihkan tatasusunan
                  // selepas panggilan rekursi.
}
```

(i) Banding dan bezakan dari segi strategi dan kecekapan algoritma-algoritma di atas.

(ii) Apa yang teijadi jika panggilan `partition` dihapuskan dari `quicksort` dan panggilan `merge` dihapuskan dari `mergesort`?

(60/100)

2. (a) Satu cara untuk mengubahsuai algoritma isihan pepohon supaya algoritma berkenaan mengisih satu senarai ke dalam tertib menurun dan bukannya tertib menaik adalah dengan memasukkan output asal ke dalam sebuah tindanan.

(i) Surih algoritma isihan pepohon yang mengisih tatasusunan 40 30 50 20 ke dalam tertib menurun seperti yang dicadangkan di atas.

(ii) Apakah kecekapan kaedah ini setelah pengubahsuaian tersebut dilakukan?

(20/100**)**

(b) (i) Senaraikan pengendalian-pengendalian yang menakrif ADT jadual bersama-sama dengan huraian ringkas bagi setiap pengendalian.

(ii) Pengendalian ADT jadual yang manakah boleh digunakan untuk memaparkan semua butir data dalam jadual? Huraikan bagaimana anda menggunakan pengendalian berkenaan untuk menjalankan tugas tersebut.

(30/100)

(c) Surihkan `HeapInsert` (penyisipan ke dalam timbunan seperti yang diberikan dalam kuliah) jika data asal adalah:

(i) 1 2 3 4 5

(ii) 1 1 1 1 1

(iii) 5 4 3 1 2

(iv) 5 4 3 2 1

(v) 2 2 2 2 2

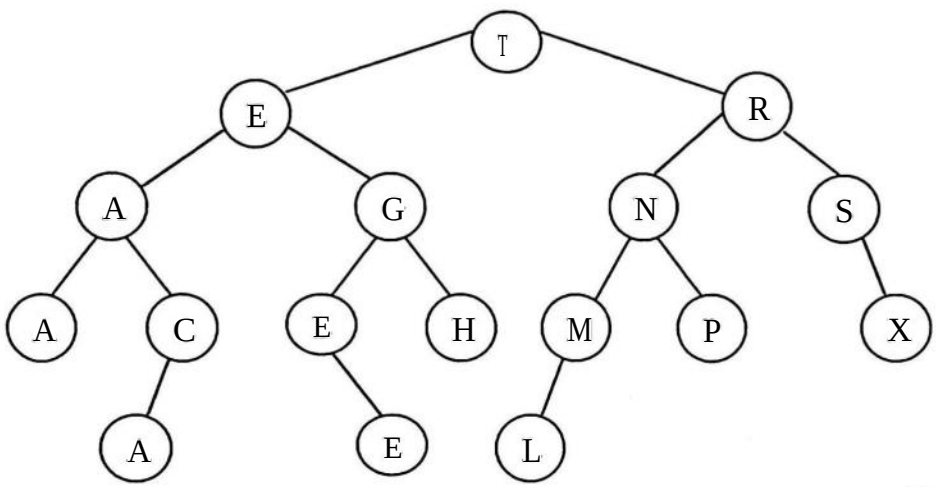
Huraikan prestasi setiap kes di atas.

(50/100)

3. (a) (i) Secara ringkas, huraikan konsep pepohon AVL.

(ii) Mengapakah kecekapan gelintaran pepohon AVL hampir secekap pepohon gelintaran perduaan dengan tinggi minimum?

(iii) Adakah pepohon di bawah merupakan sebuah pepohon AVL? Jika ya, beri sebab-sebabnya dan beri contoh satu pengendalian yang menyebabkan pepohon berkenaan menjadi bukan lagi sebuah pepohon AVL. Jika tidak, tunjuk bagaimana anda perlu melakukan putaran-putaran berkenaan supaya pepohon ini menjadi sebuah pepohon AVL.



(55/100)

(b) Berikut diberikan pseudokod untuk penyisipan nod ke dalam jadual cincangan menggunakan perantaian berasingan. Penyisipan dilakukan di depan senarai berpaut yang berkenaan.

```
TableInsert (newItem, Success)
     SearchKey = Kunci gelintaran Newltem
     I = HashIndex (SearchKey)
     P = Penuding kepada sebuah nod baru
     Setkan Success mengikut sama ada peruntukan ingatan
           berkenaan berjaya atau tidak
     if (Success)
     {     p → item = Newltem
           p → Next = T[I]
           T[I] = P
     }     // end if
```

(i) Apakah kecekapan pengendalian di atas?

(ii) Ubahsuai pseudokod di atas supaya penyisipan dilakukan di hujung senarai berpaut berkenaan.

(iii) Apakah pula kecekapan pengepdalian ini setelah diubahsuai dalam 3(b)(ii) di atas?

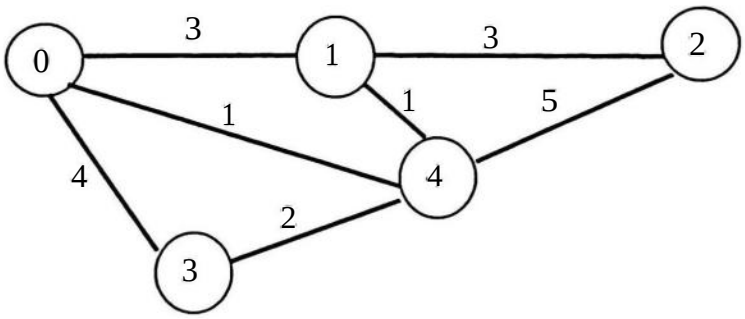
(45/100)

60

4. (a) (i) Apakah pepohan rentang minimum? Berikan satu contoh aplikasi praktikal pepohon rentang minimum.

(ii) Tanpa menulis sebarang kod, huraikan dengan menggunakan perkataan anda sendiri algoritma Prim untuk mendapatkan pepohon rentang minimum.

(iii) Lakukan algoritma Prim untuk mendapatkan pepohon rentang minimum bermula dari bucu 0 ke atas graf di bawah:



(55/100)

(b) (i) Huraikan maksud istilah-istilah berikut dalam konteks kaedah luaran: rekod data, blok, capaian blok, penimbal.

(ii) Mengapakah bilangan capaian blok perlu dikurangkan dalam pengendalian ke atas fail luaran?

(iii) Tubs fungsi (pseudokod) yang akan membaca blok-blok sebuah fail luaran secara beijujukan dan bagi setiap blok hanya rekod data terakhir sahaja diproses (dilawati) dan keseluruhan blok ditulis semula ke dalam fail berkenaan selepas pengemaskinian tersebut.

(45/100)

5. (a) (i) Huraikan kaedah penyuaian pertama dan kaedah penyuaian terbaik.

(ii) Simulasikan kedua-dua kaedah di atas ke atas timbunan yang mempunyai empat blok bebas dalam turutan berikut yang bersaiz 25 bait, 50 bait, 30 bait dan 70 bait, dan pengendalian yang dilakukan ialah peruntukan 20 bait, diikuti peruntukan 45 bait, dan seterusnya peruntukan 60 bait.

(40/100)

(b) Tanpa menulis sebarang kod huraikan dengan menggunakan ilustrasi bergrafik keadaan sebelum dan selepas pencantuman dalam algoritma `CoalesceLeft` (Pencantuman sebuah blok dengan jiran kirinya untuk membentuk blok yang lebih besar).

(20/100)

(c) Algoritma berikut mencari satu subrentetan di dalam satu rentetan yang diberikan:

```
int Find (const Strings source, const Strings target)
{
   if (source or target is empty)
      return -1;          // no match possible
   int current = 0;       // possible location in this
   while (complete match not found
                          SS any characters left in source)
      if(current character in sourcel=first target character)
            current++;
      else{do                      // found a partial match
               Step through source and target together
           while(chars left to compare SS still have a match);
           if (no more target characters to inspect)
               return current;  // found a full match,
           else current++; // keep looking
      }
   return -1; // no match found
}
```

(i) Huraikan dengan menggunakan perkataan sendiri algoritma di atas.

(ii) Ubahsuai algoritma ini supaya semua kejadian berkenaan dilokasikan dan semua kedudukan tempat kejadian berkenaan dijumpai diletakkan di dalam sebuah tindanan. Gunakan ADT tindanan.

(iii) Apakah hasil yang dijangkakan jika kandungan tindanan berkenaan dioutputkan?

(40/100)

-oooOooo -