ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐỀ THI MÔN: Cấu trúc dữ liệu và giải thuật
Lớp môn học: INT2203 1,3
Học kỳ I, Năm học 2012-2013

Thời gian làm bài: 120 phút
Đề thi và đáp án gồm 8 trang
Không sử dụng tài liệu hay thiết bị điện tử khi làm bài

Câu 1. Các phát biểu dưới đây đúng hay sai? Hãy sửa hoặc bổ sung ý nếu bạn cho là sai.

a. Chiều cao của cây tìm kiếm nhị phân n đỉnh là O(logn).
b. Với bảng băm giải quyết va chạm bằng thăm dò tuyến tính, phép remove(k) có thể thực hiện đơn giản bằng cách xóa đi giá trị ở ô tương ứng, coi như chưa bao giờ thực hiện insert vào ô đó.
c. Với bảng băm giải quyết va chạm bằng thăm dò bình phương, khi bảng chưa đầy thì phép insert luôn thực hiện được.
d. Biểu diễn đồ thị bằng ma trận kề tốt hơn biểu diễn bằng danh sách kề.
e. Dù biểu diễn đồ thị bằng ma trận kề hay bằng danh sách kề thì thời gian chạy của thuật toán đi qua đồ thị G=(V,E) theo bề rộng đều là O(|V|+|E|).
f. Luôn thực hiện được sắp xếp topo trên đồ thị có hướng không chu trình.
g. Với min heap, các phép toán findMin, findMax, deleteMin và insert đều thực hiện được trong thời gian O(logn).
h. Thuật toán thiết kế theo kỹ thuật tham ăn cho lời giải tối ưu.

Đáp án.

a. Sai. Cây TKNP lệch có độ cao là O(n).
b. Sai. Nếu chỉ xóa thì phép tìm kiếm (find) sau phép remove có thể thăm dò thiếu.
c. Sai. Xem giáo trình để rõ hơn trường hợp thăm dò bình phương không khảo sát hết bảng.
d. Sai. Ma trận kề có ưu điểm là truy cập cạnh (u, v) trong thời gian hằng. Tuy nhiên, khi đồ thị lớn, ít cạnh, ma trận kề chứa nhiều giá trị = 0 sẽ lãng phí bộ nhớ. Ngoài ra ma trận kề không hỗ trợ hiệu quả phép truy cập đến tập đỉnh kề của đỉnh u cho trước.
e. Sai. Chỉ đúng khi cài bằng danh sách kề.
f. Đúng.
g. Sai. findMin mất thời gian O(1). findMax sẽ mất thời gian O(n), ta chỉ biết min nằm ở gốc, max có thể là bất cứ đỉnh nào.
h. Sai. Tham ăn cho ta lời giải tốt, chưa chắc là tối ưu. Xem bài toán ba lô trong giáo trình.

Câu 2. Xét việc lưu tập hợp U các phần tử có giá trị khóa thuộc tập {0, 1, …, n^2-1} trong bảng băm. Với từng phương pháp giải quyết va chạm nêu dưới, một bảng băm cỡ n có thể lưu tối đa bao nhiêu khóa phân biệt?

a. Thăm dò tuyến tính
b. Thăm dò bình phương
c. Tạo dây chuyền

Đáp án.

a. n

b. n

c. n^2

Câu 3. Max heap là cây thứ tự bộ phận có tính chất *khóa cha lớn hơn khóa con*. Hãy viết giả mã thuật toán tuyến tính xây dựng max heap từ một dãy n phần tử. Vận dụng thuật toán vừa nêu cho dãy đầu vào (2203, 1, 3, 14, 16, 25, 12, 2012), hãy vẽ kết quả từng bước thực hiện.

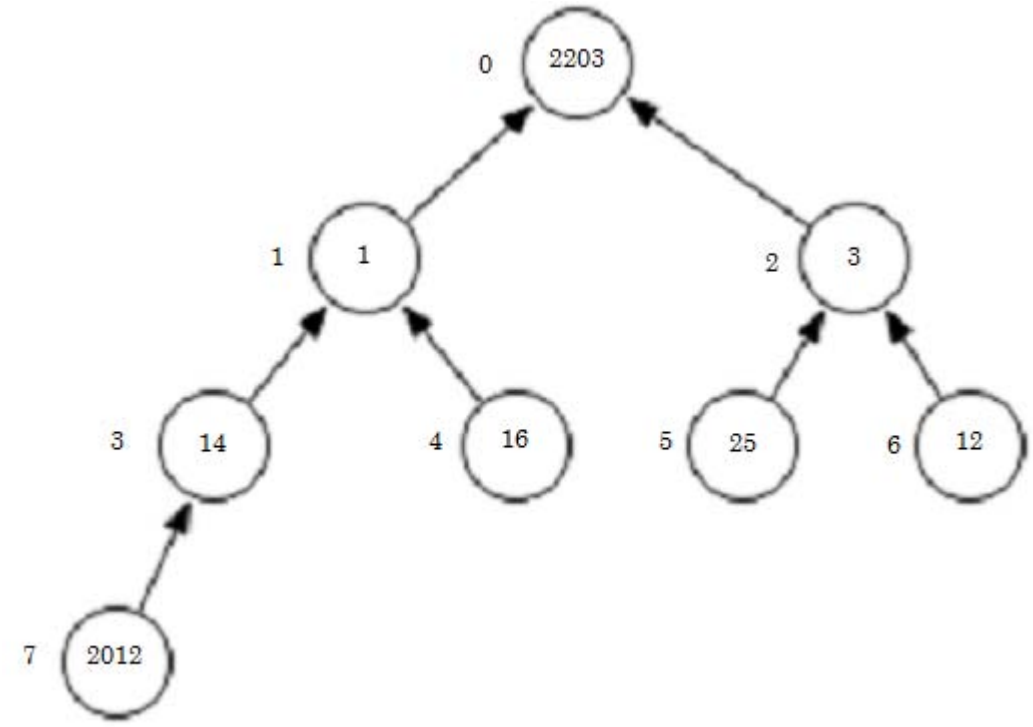
Đáp án.

- Giả mã

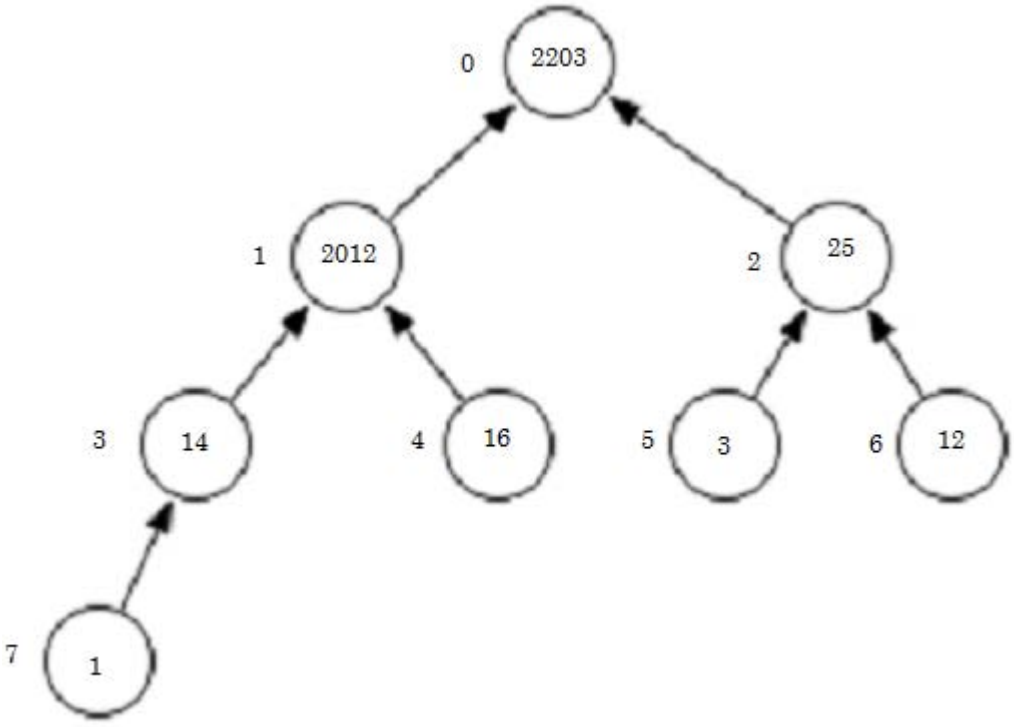
```
Algorithm BUILDHEAP(A, n)
Input: mảng A gồm n phần tử
Output: mảng A được sắp xếp lại để tương ứng với một max heap
for i ← n/2-1 to 0 do
      SIFTDOWN(A, i, n) // vận dụng thủ tục SIFTDOWN cho cây con gốc A[i]

Algorithm SIFTDOWN(A, i, n)
Input: mảng A gồm n phần tử, i ứng với chỉ số gốc của cây con cần thực hiện SIFTDOWN
Output: cây con gốc A[i] thỏa mãn tính chất max heap
parent ← i
while parent < (n – 1) / 2 do
      leftChild ← 2 x parent + 1
      rightChild ← leftChild + 1
      maxChild ← leftChild
      if rightChild < n and A[rightChild] > A[leftChild] then
            maxChild  ← leftChild
      if A[parent] < A[maxChild] then
            SWAP(A[parent] < A[maxChild])
            parent ← maxChild
      else
            break
```

- Ví dụ

  Ban đầu


0
2203
1
1
2
3
3
14
4
16
5
25
6
12
7
2012

i = 3 …
i = 2 …
i = 1 …
i = 0


0
2203
1
2012
2
25
3
14
4
16
5
3
6
12
7
1

Câu 4. Câu này hỏi về thuật toán sắp xếp nhanh lấy chốt là phần tử đầu, nhằm sắp giảm dần một dãy số thực:

a) Hãy viết mã C++ của thuật toán.

b) Với đầu vào nào thì xảy ra thời gian chạy xấu nhất. Cho ví dụ.

c) Phân tích độ phức tạp thời gian trong trường hợp xấu nhất.

Đáp án.

a)

```
void partition(double b[], int m, int& pPos){
    double pivot = b[0];
    int left = 1, right = m - 1;
    while(left <= right){
        while(left <= right && b[left] >= pivot) left++;
        while(left <= right && b[right] < pivot) right--;
        if(left < right){
            swap(b[left], b[right]);
            left++;
            right--;
        }
    }
    swap(b[0], b[right]);
    pPos = right;
}

void quicksort(double a[], int n){
    if(n <= 1) return;
    int pivotPos;
    partition(a, n, pivotPos);
    quicksort(a, pivotPos);
    quicksort(a + pivotPos + 1, n - pivotPos - 1);
}
```

b)

Trong hàm phân hoạch, khi chốt chia mảng n phần tử thành một phần không có phần tử nào, một phần chứa n-1 phần tử thì có thời gian chạy xấu nhất. Ví dụ đầu vào là dãy có thứ tự ngược với thứ tự yêu cầu (1, 3, 5, 7, 8).

c)

Ta có $T(n) = O(n) + T(0) + T(n-1) = O(n^2)$

Câu 5. Xét 6 thuật toán sắp xếp: 1-sắp xếp xen vào, 2-sắp xếp lựa chọn, 3-sắp xếp nổi bọt, 4-sắp xếp nhanh (lấy chốt là phần tử đầu), 5-sắp xếp trộn, 6-sắp xếp sử dụng heap **và** tính chất: *Nếu đầu vào là một danh sách đã sắp đúng thứ tự thì các bước của thuật toán không thực hiện bất kì biến đổi nào trên danh sách.*

Thuật toán nào có tính chất trên? Thuật toán nào không có tính chất trên? Giải thích.

Đáp án.

1, 2, 3, 4, 5 có tính chất trên
6 không có tính chất trên

Câu 6. Xét 6 thuật toán sắp xếp ở câu trước và thuật toán sắp xếp cơ số. Theo bạn, thuật toán nào tốt nhất cho mỗi mô tả sắp xếp bộ dữ liệu dưới đây? Giải thích.

a) Mảng có 32000000 phần tử nguyên trong khoảng từ 0 đến 32000000.
b) Sắp xếp độc lập 1000000 mảng, mỗi mảng có 5 phần tử.
c) Sắp xếp mảng 1000000 phần tử với thời gian chạy xấu nhất là O(nlogn).

Đáp án.

a) SX cơ số
b) SX chèn
c) SX trộn

Câu 7. Bài toán tìm xâu con chung dài nhất của một tập S các xâu được ứng dụng nhiều trong tin sinh học. Xâu con chung của 2 xâu là chuỗi các ký tự liền nhau có mặt trong cả 2 xâu. Ví dụ tập S gồm 2 xâu là "HELLO" và "ALOHA" thì xâu con chung dài nhất là "LO".

a) Hãy viết giả mã thuật toán quy hoạch động tìm xâu con chung dài nhất của 2 xâu.
b) Vẽ bảng quy hoạch động cho 2 xâu ví dụ nói trên.

Đáp án.

- Tham khảo: http://www.ics.uci.edu/~dan/class/161/notes/6/Dynamic.html
- Ý tưởng: Gọi 2 xâu là A và B, độ dài lần lượt là m và n. Bảng quy hoạch động có các hàng ứng với các ký tự trong A, cột ứng với các ký tự trong B. Ô (i, j) lưu độ dài của xâu *hậu tố* chung dài nhất của i ký tự đầu của A và j ký tự đầu của B.
    - Các ô ở hàng 0, cột 0 có giá trị = 0: L(0, j) = L(i,0) = 0
    - L(i,j) = 1 + L(i – 1, j – 1) nếu A[i] = B[j]
    - L(i,j) = 0 nếu A[i] ≠ B[j]

a) Giả mã

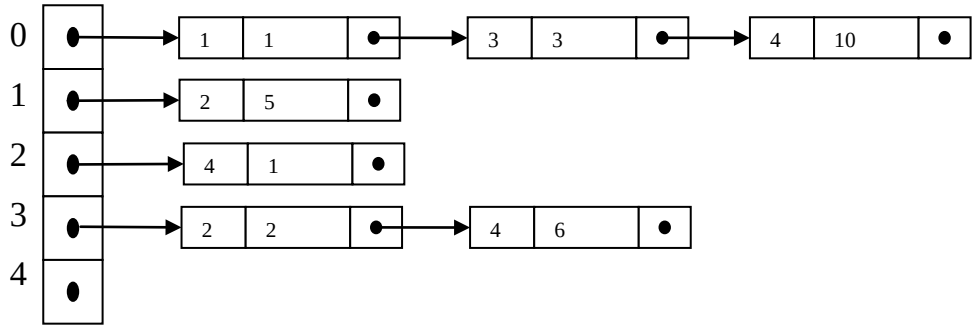
```
Algorithm LCSTR(A, m, B, n)
Input: xâu A chiều dài m, xâu B chiều dài n
Output:
   + chiều dài xâu con chung dài nhất của A và B lưu trong maxLen
   + vị trí ký tự cuối của xâu cần tìm trong A lưu trong i
   + vị trí ký tự cuối của xâu cần tìm trong B lưu trong j
for i ← 0 to m do L(i, 0) ← 0
for j ← 0 to n do L(0, j) ← 0
maxLen ← 0
answer ← <0, 0>
for i ← 1 to m do
     for j ← 1 to n do
```

```
if A[i] ≠ B[j] then L(i, j) ← 0
else
     L(i, j) ← 1 + L(i-1, j-1)
     if L(i, j) > maxLen then
          maxLen ← L(i,j)
          answer ← <i, j>
```

b) Bảng quy hoạch động

		A	L	O	H	A
	H	0	0	0	1	0
	E	0	0	0	0	0
	L	0	1	0	0	0
	L	0	1	0	0	0
	O	0	0	2	0	0

Câu 8. Đồ thị có hướng có trọng số G được cho trong danh sách kề ở hình bên dưới. Mỗi nút trong danh sách liên kết có 3 thành phần: số hiệu đỉnh, trọng số cung và địa chỉ nút tiếp theo.



Hãy

a) Vẽ đồ thị G.

b) Cho biết G liên thông mạnh, liên thông yếu hay không liên thông? Giải thích ngắn gọn.

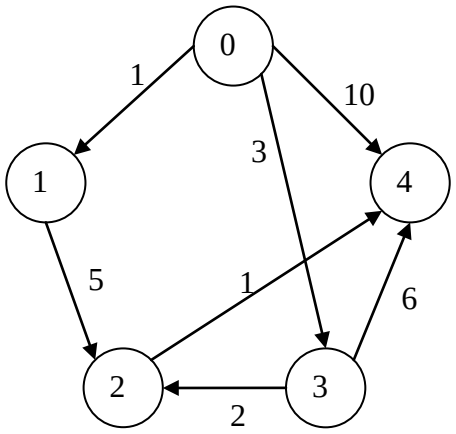
c) Cho biết 1 kết quả của thuật toán Tarjan sắp xếp topo trên G và các bước dẫn tới kết quả này.

d) Cho biết kết quả và các bước thực hiện thuật toán Dijkstra tìm độ dài đường đi ngắn nhất từ đỉnh 0 tới các đỉnh còn lại.

e) Cho biết kết quả và các bước thực hiện thuật toán Prim tìm cây khung nhỏ nhất của đồ thị vô hướng nền của G.

Đáp án.

a)



b) G không liên thông mạnh vì có đường đi xuất phát từ đỉnh 0 nhưng không có đường đi tới đỉnh 0.

G liên thông yếu vì đồ thị vô hướng nền của nó là liên thông.

c) L = (0, 3, 1, 2, 4)

Các bước:

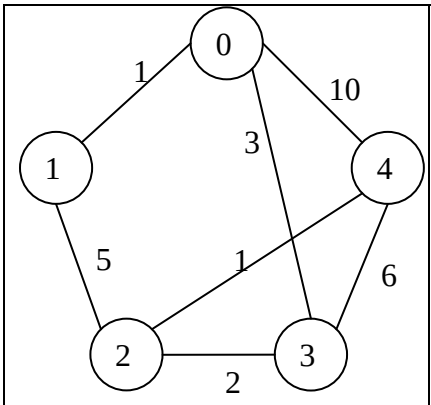
```
L = ()
DFS(0)
      DFS(1)
            DFS(2)
                  DFS(4)
                        L = (4)
                  L = (4, 2)
            L = (4, 2, 1)
      DFS(3)
            L = (4, 2, 1, 3)
      L = (4, 2, 1, 3, 0)
L = (0, 3, 1, 2, 4) // đảo ngược
```

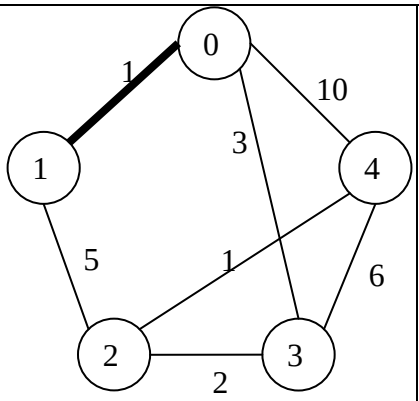
d)

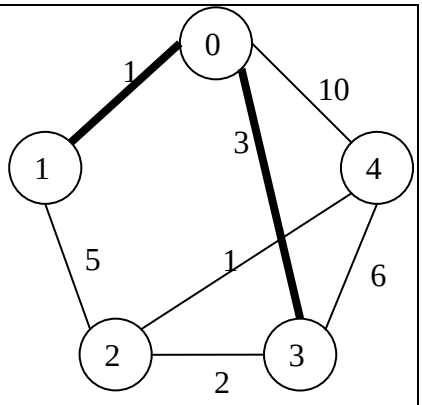
	S	D[1]	D[2]	D[3]	D[4]
Khởi tạo	{0}	1	∞	3	10
Thêm 1 vào S	{0, 1}		min{∞, D[1] + c(1, 2)} = 6	min{3, D[1] + c(1, 3)} = 3	min{10, D[1] + c(1,4)} = 10
Thêm 3 vào S	{0, 1, 3}		min{6, D[3] + c(3, 2)} = 5		min{10, D[3] + c(3, 4)} = 9
Thêm 2 vào S	{0, 1, 3, 2}				min{9, D[2] + c(2, 4)} = 6
Thêm 4 vào S	{0, 1, 3, 2, 4}				

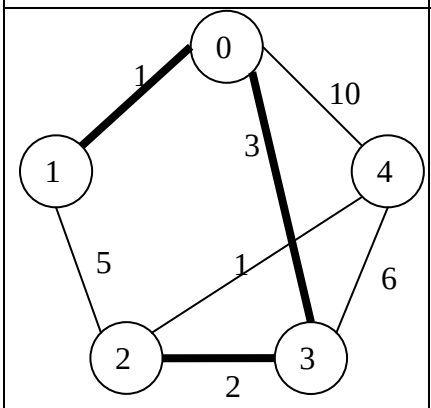
Độ dài đường đi ngắn nhất từ 0 đến v được lưu trong ô D[v]

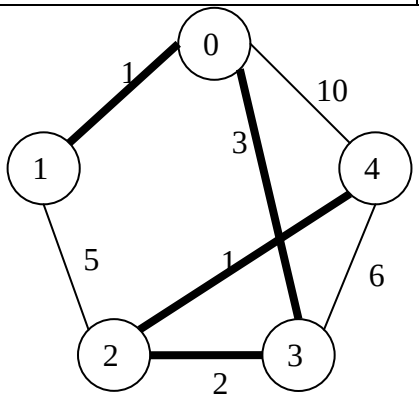
e)

<table>
<tr>
<td> 

T = {}; U = {0}</td>
<td> 

T = T ∪ (0, 1); U = {0, 1}</td>
<td> 

T = T ∪ (0, 3); U = {0, 1, 3}</td>
</tr>
<tr>
<td> 

T = T ∪ (3, 2);
U = {0, 1, 3, 2}</td>
<td colspan="2"> 

T = T ∪ (2, 4);
U = {0, 1, 3, 2, 4}
Cây khung nhỏ nhất này có độ dài = 7</td>
</tr>
</table>