|  |  |  |
| --- | --- | --- |
| TRƯỜNG ĐẠI HỌC NHA TRANG  **KHOA CÔNG NGHỆ THÔNG TIN**  Người ra đề  Nguyễn Đức Thuần | ĐỀ THI KẾT THÚC MÔN HỌC  MÔN: **CẤU TRÚC DỮ LIỆU**  Thời gian: 60 phút  Lớp **61TH**  *Không sử dụng tài liệu*  **ĐỀ I**  **Ngày thi: 13 /01/2021** | DUYỆT ĐỀ  Nguyễn Đức Thuần |

**Câu I**: (3.0đ)

1. (1.0đ) Chuyển đổi biểu thức trung tố sau sang b.thức hậu tố:

3\*((4\*3/7)-5/2)-(4/3\*5-7)

1. (1.5đ) Viết hàm tìm kiếm nhị phân trên dãy có n phần tử đã được sắp. Đánh giá độ phức tạp của hàm vừa trình bày.
2. (0.5đ) Cho biết độ phức tạp của thuật toán Quick-Sort.

**Câu II**: (2.0đ)

1. Xây dựng cây AVL lần lượt từ các giá trị sau: 25, 30, 20, 10, 5, 15, 12
2. Xóa nút 20 từ cây AVL đã xây dựng ở câu a và cân cân chỉnh lại cây để cây là cây AVL.

**Câu III**: (5.0đ) Cây nhị phân.

Thực hiện các công việc sau:

1. (0.5đ) Khai báo dữ liệu để quản lý các nút thuộc một cây nhị phân tìm kiếm, chứa các giá trị là số nguyên.
2. (1đ ) Viết thủ tục thực hiện tạo một cây nhị phân tìm kiếm từ các số nguyên nhập từ bàn phím, kết thúc việc nhập khi gõ 0.
3. (0.5đ) Viết thủ tục duyệt cây theo thứ tự LNR.
4. (1đ) Viết hàm xác định chiều cao của cây.
5. (1đ) Viết thủ tục in ra các nút ở mức i.
6. (1đ) Viết thủ tục sao chép cây nhị phân tìm kiếm có gốc được trỏ bởi biến **goc1**, thành cây có gốc được trỏ bởi biến **goc2**.

|  |  |  |
| --- | --- | --- |
| TRƯỜNG ĐẠI HỌC NHA TRANG  **KHOA CÔNG NGHỆ THÔNG TIN**  Người ra đề  Nguyễn Đức Thuần | ĐỀ THI KẾT THÚC MÔN HỌC  MÔN: **CẤU TRÚC DỮ LIỆU**  Thời gian: 60 phút  Lớp **61TH**  *Không sử dụng tài liệu*  **ĐỀ II**  **Ngày thi: 13 /01/2021** | DUYỆT ĐỀ  Nguyễn Đức Thuần |

**Câu I**: (3.0đ)

1. (1.0đ) Chuyển đổi biểu thức trung tố sau sang biểu thức hậu tố:

(4/5\*3-4/7)\*4-6\*(5/4-4\*3)

1. (1.5đ) Trình bày các bước sắp xếp dãy số sau bằng thuật toán Quick Sort:

14 23 10 56 8 30 5 28 78 42 25 50

1. (0.5đ) Cho biết độ phức tạp của thuật toán Heap Sort.

**Câu II**: (2.0đ)

1. Xây dựng cây AVL lần lượt từ các giá trị sau: 30, 20, 25, 50, 60, 10, 23, 21
2. Xóa nút 25 từ cây AVL đã xây dựng ở câu a và cân cân chỉnh lại cây để cây là cây AVL.

**Câu III**: (5.0đ) Danh sách liên kết

Thực hiện các công việc sau:

1. (0.5đ) Khai báo cấu trúc để quản lý các nút chứa các số nguyên bằng danh sách liên kết đơn.
2. (1đ) Tạo một danh sách liên kết từ các số nguyên nhập từ bàn phím, kết thúc việc nhập khi gõ 0.
3. (0.5đ) Viết thủ tục duyệt danh sách.
4. (1đ) Viết thủ tục sắp xếp các nút theo thứ tự tăng dần.
5. (1đ) Viết thủ tục đảo ngược danh sách không được tạo danh sách mới.
6. (1đ) Viết thủ tục xóa các nút trùng nhau trong một danh sách liên kết đã được sắp. (các nút trùng nhau chỉ giữ lại 1 nút)

ĐÁP ÁN

**ĐỀ I**

**Câu I**: (3đ)

|  |  |  |
| --- | --- | --- |
| a. | 3 4 3 \* 7 / 5 2 / - \* 4 3 / 5 \* 7 - - | 1đ |
| b. | int Timnp(float a[], int l, int r, float X)  { int k;  if (l>r)  return 0;  else  { k=(l+r)/2;  if (a[k]==X) return k;  else  if (a[k]>X) return Timnp(a,l,k-1);  else return Timnp(a,k+1,r);  } | 0.75đ |
|  | T(n)= O(log2n) | 0.75đ |
| c | Thuật toán Quick-Sort: Tốt nhất O(nlog2n); Xấu nhất O(n2) | 0.5đ |

**Câu II**: (2đ)

|  |  |  |
| --- | --- | --- |
|  | Thực hiện phép quay đơn tại nút 20 | 0.5đ |
|  | Thực hiện phép quay kép tại nút 25 (lần lượt tại nút 10, 25) | 1đ |
|  | Chép nút 15 lên nút 25, nối phải nút 10 là nút 12, xóa nút 15.  Không cần cân chỉnh lại cây. | 0.5đ |

**Câu III** (5đ)

|  |  |  |
| --- | --- | --- |
|  | struct nut {  int Info;  nut \*left,\*right;  };  typedef nut Node; | 0.5đ |
|  | void mocnut (Node \*&goc,int x)  { if (goc==NULL)  { goc=new Node;  goc->Info=x;  goc->left=NULL;  goc->right=NULL;  }  else  if (goc->Info>=x)  mocnut(goc->left,x);  else  mocnut(goc->right,x);  }  //-------------------------------------------------  void taocay(Node \*&goc)  { int tam;  do{  printf (" Nhap 1 so nguyen, 0:dung ");  scanf("%d",&tam);  if (tam!=0)  mocnut(goc,tam);  }  while (tam!=0);  } | 1đ |
|  | void duyetLNR(Node \*goc)  { if (goc!=NULL)  { duyetLNR(goc->left);  printf("%3d",goc->Info);  duyetLNR(goc->right);  }} | 0.5đ |
|  | int max(int a,int b)  { if (a>=b)  return a;  else  return b;  }  //-------------------------------------------------  int chieucao(Node \*goc)  { if (goc==NULL)  return 0;  else  if ((goc->left==NULL)&&(goc->right==NULL))  return 1;  else  return 1+max(chieucao(goc->left),chieucao(goc->right));  } | 1đ |
|  | void Inmuc(Node \*goc,int i,int m)  { if (goc!=NULL)  if (m==i)  printf("%3d",goc->Info);  else  { Inmuc(goc->left,i,m+1);  Inmuc(goc->right,i,m+1);  }  } | 1đ |
|  | void chepcay(Node \*goc,Node \*&goc1)  { if (goc!=NULL)  { mocnut(goc1,goc->Info);  chepcay(goc->left,goc1->left);  chepcay(goc->right,goc1->right);  }  } | 1đ |

ĐÁP ÁN

**ĐỀ II**

**Câu I:** (3đ)

|  |  |  |
| --- | --- | --- |
|  | 4 5 / 3 \* 4 7 / - 4 \* 6 5 4 / 4 3 \* - \* - | 1đ |
|  |  | 1.5đ |
|  | T(n)=O(nlog2n) | 0.5đ |

**Câu II**: (2đ)

|  |  |  |
| --- | --- | --- |
|  | Thực hiện quay kép tại nút 30 (lần lượt tại các nút 20, 30) | 1đ |
|  | Thực hiện quay đơn tại nút 30 | 0.5đ |
|  | Chép nút 23 lên nút 25, nối phải nút 20 là nút 21, xóa nút 23.  Không cần cân chỉnh lại cây. | 0.5đ |

**Câu III:** (5đ)

|  |  |  |
| --- | --- | --- |
|  | struct nut {  int gtri;  nut \*tiep; };  typedef nut NODE; | 0.5đ |
|  | void nhapdl(NODE \*&dau){  NODE \*p,\*q;  int tam;  do  { printf("nhap vao 1 so nguyen (0:dung)");  scanf("%d",&tam);  if(tam!=0)  { p=new NODE;  p->gtri=tam;  p->tiep=NULL;  // Moc nut vao ds  if (dau==NULL)dau=p;  else  q->tiep=p;  q=p;  } }  while (tam!=0); } | 1đ |
|  | void duyetds(NODE \*dau){  if(dau!=NULL)  {  printf("%3d",dau->gtri);  duyetds(dau->tiep);  } } | 0.5đ |
|  | void sapxep(NODE \*&dau)  {  NODE \*p,\*q;  int tam;  p=dau;  while (p->tiep!=NULL)  {q=p->tiep;  while(q!=NULL)  {  if(q->gtri<p->gtri){  tam=p->gtri;  p->gtri=q->gtri;  q->gtri=tam;  }  q=q->tiep; }  p=p->tiep; }  } | 1đ |
|  | void daonguoc(NODE \*&dau)  { NODE \*p, \*q;  q=NULL;  p=dau;  while (p!=NULL)  {dau=dau->tiep;  p->tiep=q;  q=p;  p=dau; }  dau=q;  } | 1đ |
|  | void xoatrung(NODE \*&dau)  {NODE \*p,\*q;  p=dau;  while (p->tiep!=NULL)  {q=p->tiep;  if (p->gtri==q->gtri)  { p->tiep=q->tiep;  delete q;  }  else  p=p->tiep;  } } | 1đ |