**PHẦN 1**

**Câu 1:** **TT1:** Ta thấy phép so sánh n==0 là PTTC của tt.

Gọi g(n) là số lần thực hiện PTTC của CTC kích thước n. g(n/2) là … n/2.

Khi n=0, g(n)=g(0)=1. Khi n>0 thì pt gọi lại CTC HamMu(x,n/2) 2 lần

* g(n)=

Ta có g(n)=2+1=2[2]+1=4=8=…=2k+2k-1 (1)

Kích thước chương trình giảm xuống thì k sẽ tang lên nên chương trình sẽ dừng khi

n/2k=1⬄ n=2k ⬄ k=log2n =>(1) ⬄ g(n)=n.g(1)+n-1=n(2g(0)+1)+n-1=4n-1

Vậy độ phức tạp của thuật toán này làT(n)=O(4n-1)= O(n)

**TT2:** Ta thấy phép so sánh n==0 là PTTC của tt.

Gọi g(n) là số lần thực hiện PTTC của CTC kích thước n. g(n/2) là … n/2.

Khi n=0, g(n)=g(0)=1. Khi n>0 thì pt gọi lại CTC HamMu(x,n/2) 1 lần

* g(n)=

Ta có g(n)=+1=[]+1==…=+k (1)

Kích thước chương trình giảm xuống thì i sẽ tang lên nên chương trình sẽ dừng khi

n/2k=1⬄ n=2k ⬄ k=log2n =>(1) ⬄ g(n)=g(1)+ log2n =g(0)+1+ log2n = log2n+2

Vậy độ phức tạp của thuật toán này là T(n)=O(log2n+2)= O(log2n)

**Câu 2:** **TT1:** Thuật toán dùng 1 vòng lặp với kích thước n. Sử dụng quy tắc tổng : O(1)+O(n-1)+O(1)+O(1) = O(n)

**TT2:** Gọi g(n) là số lần thực hiện của PTTC (l==r) của hàm Max(A,0,n-1)

Gọi (n/2) là số lần thực hiện của PTTC của CTC kích thước n-2

Khi n=1, g(n)=g(1)=1. Khi n=2, g(n)=g(2)=1. Khi n>2, CT gọi lại CTC 2 lần.

* g(n)=

Ta có g(n)=2+1=2[2]+1=4=8=…=2k+2k-1 (1)

Kích thước chương trình giảm xuống thì k sẽ tang lên nên chương trình sẽ dừng khi

n/2k=1 hoặc n/2k=2. Ta có trường hợp xấu nhất là n/2k=1 ⬄ k=log2n

* g(n)=ng(1)+n-1=2n-1. Vậy độ phức tạp của thuật toán này làT(n)=O(2n-1)= O(n)

**Câu 3:**

**a.** int DemSoCapNghichDao(int A[], int n)

{ int i,j; int dem=0;

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

if(i<j&&A[i]>A[j])

dem++;

return dem;

}

**b.** int DemSoCapBangX(int A[], int n, int x)

{ int i,j; int dem=0;

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

if(A[i]+A[j]==x)

dem++;

return dem;

}

**Câu 4**

|  |  |
| --- | --- |
| **a.** void DaoNguocLap(int A[],int Left,int Right)  { while(Left<Right){  int tam=A[Left];  A[Left]=A[Right];  A[Right]=tam;  Left++; Right--;  }  } | **b.** void DaoNguocDQ(int A[],int Left,int Right)  { if(Left<Right){  int tam=A[Left];  A[Left]=A[Right];  A[Right]=tam;  DaoNguocDQ(A,Left+1,Right-1);  }  } |

**Câu 5** + Hàm fun1(x,y) tính và trả về giá trị của biểu thức: ((1+2+…+(x-1)+x)+y). Với lời gọi hàm fun1(5,2) ta có kết quả sau : ((1+2+3+4+5)+2)=17

+ Hàm fun2(x,y) tính và trả về giá trị của biểu thức: x\*y. Với lời gọi hàm fun2(4,3), ta có kết quả sau: (4\*3)=12

**Câu 6 a.** unsigned int f(unsigned int n)

{ if(n==1) return 1;

else return f(n-1)+2\*n;

}

**b.** Ta có f(5)=29 vì : hàm f(5) gọi f(4), f(4) gọi f(3), f(3) gọi f(2), f(2) gọi f(1) và ta có f(1)=1

Suy ngược, ta sẽ có f(2)=1+2\*2=5, f(3)=5+2\*3=11, f(4)=11+2\*4=19, f(5)=19+2\*5=29

**Câu 7**  void InDS(int A[], int ds[], int n)

{ int i,j;

for(i=0;i<n;i++)

ds[i]=0;

for(i=0;i<n;i++){

for(j=i+1;j<n;j++){

if(A[j]<A[i])

ds[i]++;

}

}

for(i=0;i<n;i++)

printf("%d\t",ds[i]);

}

**Câu 8**

|  |  |
| --- | --- |
| int gcd(int n, int m)  { if((m<=n)&&(n%m==0))  return m;  else if(n<m)  return gcd(m,n);  else if((m<n)&&(n%m!=0))  return gcd(m,n%m);  } | int gcdLap(int a, int b)  {  int c;  while(a!=0){  c=a; a=b%a; b=c;  }  return b;  } |

**Câu 9**

|  |  |
| --- | --- |
| TT1: Ta thấy PTTC là : p=p\*x/j;  Khi i=1, số lần thực hiện của PTTC là n  Khi i=2, số lần thực hiện của PTTC là n-1  Khi i=n, số lần thực hiện của PTTC là 1  Tổng số lần thực hiện là 1+…+n=n(n-1)/2  Vậy, T(n)=O(1/2\*(n2-n))=O(n2-n)=O(n2) | TT2: Ta thấy PTTC là : p=p\*x/i  Thuật toán dùng 1 vòng lặp với kích thước n nên số lần thực hiện của PTTC là n  Vậy, T(n)=O(n)  TT2 tối ưu hơn vì nó có ĐPT thấp hơn nên thời gian thực hiện sẽ nhanh hơn. |

**Câu 10**

|  |  |
| --- | --- |
| int KTDoiXung(int A[], int n)  { int i=0; int j=n-1;  while(i<j)  { if(A[i]!=A[j]) return 0;  else { i++; j--;}  }  return 1;  } | intKTDoiXungDQ(int A[], int L, int R)  { if(((L==0)&&(R==0))||(L==R)) return 1;  else if(A[L]!=A[R])  return 0;  else  return KTDoiXungDQ(A,L+1,R-1);  } |

**PHẦN 2**

**Câu 1**

**a.** **NutMH \*Addr(NutMH \*FirstS, int BMaMH)**

{ if(FirstS==NULL) return NULL;

else if(FirstS->MaMH==BMaMH)

return FirstS;

else return Addr(FirstS->Next, BMaMH);

}

**b.** **void ListOfSubjects(NutGV \*FirstT, St8 BMaGV)**

{ if((FirstT!=NULL)&& strcmp(FirstT->MaGV,BMaGV)<=0)

{ if(strcmp(FirstT->MaGV,BMaGV)==0)

LOS(FirstT->DSM);

else ListOfSubjects(FirstT->Down,BMaGV);

}

}

**void LOS(NutM \*F)**

{ if(F!=NULL)

{ int stt=1;

cout<<"STT Ten mon hoc So don vi hoc trinh"<<endl;

while(F!=NULL)

{ NutMH \*p=Addr(FirstS,F->MaM);

cout<<stt<<p->TenMH<<p->SoDVHT<<endl;

F=F->Tiep; stt=stt+1;

}

}

}

**Câu 2**

**a.** **float TongDoanhThu(ThanhPho \*First, St25 Bthanhpho)**

{ if((First==NULL)||strcmp(First->TenThanhPho, Bthanhpho)>0)

return 0;

else if(strcmp(First->TenThanhPho, Bthanhpho)==0)

return Tong(First->DSDaily);

else return TongDoanhThu(First->Down, Bthanhpho);

}

**float Tong(DaiLy \*F)**

{ if(F==NULL) return 0;

else return F->DoanhThu + Tong(F->Tiep);

}

**b.** **void InDSDaiLy(ThanhPho \*First)**

{ if(First!=NULL)

{ cout<<"DANH SACH CAC DAI LY"<<endl;

int stt=1;

while(First!=NULL)

{ cout<<stt<<"."<<First->TenThanhPho<<":";

DaiLy \*p=First->DaiLy;

cout<<p->DSDaiLy<<"("<<p->SDT<<")";

p=p->Tiep;

while p!=NULL do

{ cout<<","<<p->DSDaiLy<<"("<<p->SDT<<")";

p=p->tiep;

}cout<<endl; First=First->Down; stt=stt+1;

}

}

}

**Câu 3 a.** **void DSDau(SinhVien \*F, St20 Dep)**

{ while(F!=NULL && strcmp(F->Khoa,Dep)<0)

F=F->Next;

while(F!=NULL && strcmp(F->Khoa,Dep)==0)

{ if(F->D1>=5 && F->D2>=5 && F->D3>=5)

cout<<F->Hoten<<endl;

F=F->next;

}

}

**b.** **void Bosung(SinhVien \*&F, St20 Name, St20 Dep)**

{ if((F==NULL)||strcmp(F->Khoa,Dep)>0||

(strcmp(F->Khoa,Dep)==0 && strcmp(F->HoTen,Name)>0))

{ SinhVien \*P; P= new SinhVien;

strcpy(P->HoTen,Name); strcpy(P->Khoa,Dep);

P->Next=F; F=P;

}

else if(strcmp(F->Khoa,Dep)<0||

(strcmp(F->Khoa,Dep)==0 && strcmp(F->HoTen,Name)<0))

Bosung(F->Next,Name,Dep);

}

**Câu 4 a.** **Ham \*Addr(Ham \*F, St10 TH)**

{ if(F==NULL || strcmp(F->Name,TH)>0)

return NULL;

else if(strcmp(F->Name,TH)==0)

return F;

else return Addr(F->next,TH);

}

**b.** **void See(Ham \*F, St10 TH)**

{ Ham \*Q=Addr(F,TH);

if(Q!=NULL)

{ cout<<Q->Description<<endl;

HLQ \*P=Q->SeeAlso;

while(P!=NULL)

{ cout<<P->TenHLQ<<endl;

P=P->Tiep;

}}}

**c.** **void Remove(Ham \*&F, St10 TH)**

{ if(F!=NULL)

if(strcmp(F->Name,TH)==0)

{ XoaCacHLQ(F); Ham \*Q=F;

F=F->Next; delete(Q);

}

else if(strcmp(F->Name,TH)<0)

Remove(F->Next.TH);

}

**void XoaHLQ(HLQ \*&L, St10 X)**

{ if(L!=NULL)

if(strcmp(L->TenHLQ,X)==0)

{ HLQ \*P=L; L=L->Tiep; delete(P);

}

else XoaHLQ(L->Tiep,X);

}

**void XoaCacHLQ(Ham \*F)**

{ while(F->SeeAlso!=NULL) do

{ St10 X; strcpy(X,F->SeeAlso->TenHLQ);

XoaHLQ(F->SeeAlso,X); Ham \*Q=Addr(F,X);

XoaHLQ(Q->SeeAlso,F->Name);

}

}

**Câu 5 a.** **void XoaDL(DaiLy \*&F, St25 Bdaily)**

{ if(F!=NULL)

if(strcmp(F->TenDaiLy,Bdaily)==0)

{ DaiLy \*q=F; F=F->Tiep; delete(q); }

else XoaDL(F->Tiep,Bdaily);

}

**b.** **void Xoa(ThanhPho \*&First, St25 Bthanhpho, St25 Bdaily)**

{ if(First!=NULL)

{ if(strcmp(First->TenThanhPho,Bthanhpho)==0)

{ XoaDL(First->DSDaily,Bdaily);

if(First->DSDaily==NULL)

{ ThanhPho \*p=First;

First=First->Down;

delete(p);

}

}

else if(strcmp(First->TenThanhPho,Bthanhpho)<0)

Xoa(First->Down,Bthanhpho,Bdaily);

}

}

**Câu 6 a.** **NutGV \*Addr(NutGV \*FirstT, St8 BMaGV)**

{ if (FirstT == NULL || strcmp(FirstT->MaGV, BMaGV) > 0)

return NULL;

else if (strcmp(FirstT->MaGV, BMaGV) == 0)

return FirstT;

else return Addr(FirstT->down, BMaGV);

}

**b.** **void ListOfTeachers(NutGV \*FirstT, int BMaGV)**

{ if (FirstT != NULL)

{ NutM \*P = FirstT->DSM;

while (P != NULL && P->MaM != BMaM)

P = P->Tiep;

if (P != NULL)

cout << FirstT->HoTenGV << endl;

ListOfTeachers(FirstT->down, BMaM);

}

}

**Câu 7 a.** **void Push(Nut \*&Top, int X)**

{ Nut \*p; p=new Nut; p->Info=X; p->Next=Top; Top=p; }

**int Pop(Nut \*&Top)**

{ if (Top != NULL)

{ int X=Top->Info; Nut \*p=Top; Top=p->Next; delete(p); return X; }

}

**b.** **void LKeNguoc(Nut \*First)**

{ Nut \*p = First; Nut \*Top = NULL;

while (p != NULL)

{ Push(Top, p->Info); p = p->Next; }

while (Top != NULL) cout << Pop(Top) << endl;

}

**c.** **void LKeNguoc(Nut \*First)**

{ if (First != NULL)

{ LKeNguoc(First->Next); cout << First->Info << endl;}

}

**Câu 8 a.** **Ham \*Addr(Ham \*F, St10 TH)**

{

if (F == NULL||strcmp(F->Name, TH)>0) return NULL;

if (strcmp(F->Name, TH) == 0) return F;

return Addr(F->next, TH);

}

**b.** **void BSCuoi(HLQ \*&L, St10 X)**

{ if (L == NULL)

{ L = new HLQ; strcpy(L->TenHLQ, X); L->Tiep = NULL; }

else if (strcmp(L->TenHLQ, X) != 0) BSCuoi(L->Tiep, X);

}

**c.** **void Add\_SeeAlso(Ham \*F, St10 TH1, St10 TH2)**

{ Ham \*Q1 = Addr(F, TH1); Ham \*Q2 = Addr(F, TH2);

if ((Q1 != NULL) && (Q2 != NULL))

{ BSCuoi(Q1->SeeAlso,TH2); BSCuoi(Q2->SeeAlso,TH1);}

}

**Câu 9 a.** **void Copy(Nut \*L, Nut \*&G)**

{if (L == NULL) G = NULL;

else { G = new Nut; G->So = L->So;Copy(L - Tiep, G->Tiep);}

}

**b.** **void Hop(Nut \*L1, Nut \*L2, Nut \*&L3)**

{if (L1 == NULL) Copy(L2, L3);

else{ if (L1->So = L2->So)

{L3 = new Nut; L3->So = L1->So;

Hop(L1 - Tiep, L2->Tiep, L3->Tiep);

};

if (L1->So < L2->So)

{L3 = new Nut; L3->So = L1->So;

Hop(L1->Tiep, L2, L3->Tiep);

};

if (L1->So > L2->So)

{L3 = new Nut; L3->So = L2->So;

Hop(L1, L2->Tiep, L3->Tiep);

};

}

}

**Câu 10 a.** **int KT(Nut \*L)**

{if (L == NULL || L->Tiep == NULL) return 1;

else return (L->So < L->Tiep->So) && KT(L->Tiep);

}

**b.** **void Giao(Nut \*L1, Nut \*L2, Nut \*&L3)**

{if (L1 == NULL || L2 == NULL) L3 = NULL;

else{

if (L1->So = L2->So)

{L3 = new Nut; L3->So = L1->So;

Giao(L1->Tiep, L2->Tiep, L3->Tiep);

};

else{

if (L1->So < L2->So) Giao(L1->Tiep, L2, L3);

else Giao(L1, L2->Tiep, L3);

};

}

}

**PHẦN 3**

**Câu 1 a.** **CB \*DinhViCB(CB \*T, St20 Name)**

{if(T==NULL) return NULL;

else if(T->Ten==Name) return Ten;

else if(T->Ten>Name) return DinhViCB(T->Left,Name);

else return DinhViCB(T->Right,Name);

**b.** **CapNhat (CB \*T, St20 Name, St20 FL, char Level)**

{ CB \*p=DinhViCB(T,Name);

if(p==NULL) cout<<"Khong co can bo nay!";

else

{ NN \*q=q->DSNN;

while(q!=NULL)

{ if(strcmp(q->NgoaiNgu,FL)==0)

if(q->CapDo<Level)

{ q->CapDo=Level; return;}

else

{cout<<"Khong cap nhat!";return;}

else q=q->Next;

}

q = new NN; strcpy(q->NgoaiNgu,FL);

q->CapDo=Level; q->Next=p->DSNN;

p->DSNN=q;

}

}

**Câu 2 a.void Xoa(Nut \*T)**

{ if(T!=NULL)

{ Xoa(T->Right); Xoa(T->Left); delete(T); T=NULL;}

}

**b.** **int LaDong(Nut \*T)**

{ if((T==NULL)||(T->Left==NULL&&T->Right==NULL))

return 1;

else if(T->Left==NULL)

return (T->Info>T->Right->Info)&&LaDong(T->Right);

else if(T->Right==NULL)

return (T->Info>T->Left->Info)&&LaDong(T->Left);

else return (T->Info>T->Left->Info)&&LaDong(T->Left)

&&(T->Info>T->Right->Info)&&LaDong(T->Right);

}

**Câu 3 a. int TinhGT(Nut \*E)**

{switch(E->Kyhieu)

{case 'T':return 1; break;

case 'F':return 0; break;

case 'A':return TinhGT(E->Left)&&TinhGT(E->Right); break;

case 'O':return TinhGT(E->Left)||TinhGT(E->Right); break;

case 'N':return !TinhGT(E->Right); break;

}}

**b.void InBT(Nut \*E)**

{ switch(E->Kyhieu)

{case 'T':cout<<"true"; break;

case 'F':cout<<"false"; break;

case 'A':cout<<"("<<InBT(E->Left)<<") and ("<<InBT(E->Right)<<")"; break;

case 'O':cout<<"("<<InBT(E->Left)<<") or ("<<InBT(E->Right)<<")"; break;

case 'N':cout<<"not ("<<InBT(E->Right)<<")"; break;

}

}

**Câu 4 a.** **int SoNutLa(Nut \*T)**

{ if(T==NULL) return 0;

else if(T->Left==NULL&&T->Right==NULL) return 1;

else return SoNutLa(T->Left)+SoNutLa(T->Right);

}

**b.** **void DieuChinh(Nut \*T)**

{ if((T!=NULL)&&(T->Left!=NULL||T->Right!=NULL))

if(T->Left==NULL)

{ DieuChinh(T->Right);

if(T->Info<T->Right->Info)

swap(T->Info,T->Right->Info); DieuChinh(T->Right);

}

else if(T->Right==NULL)

{ DieuChinh(T->Left);

if(T->Info<T->Left->Info)

swap(T->Info,T->Left->Info); DieuChinh(T->Left);

}

else{ DieuChinh(T->Left); DieuChinh(T->Right);

if(T->Left->Info>T->Right->Info)

{ if(T->Info<T->Left->Info)

{ swap(T->Info,T->Left->Info); DieuChinh(T->Left);}

}

else

{if(T->Info<T->Right->Info)

{swap(T->Info,T->Right->Info);DieuChinh(T->Right);}

}

}

}

**Câu 5 a.** **int ChieuCao(Nut \*T)**

{ if(T==NULL) return 0;

else return Max(ChieuCao(T->Right),ChieuCao(T->Left))+1;

}

**Max(int a, int b)**

{ if(a>b) return a; else return b; }

**b.** **Nut \*Cha(Nut \*T, Nut \*p)**

{ if((T==NULL)||(T==p)) return NULL;

else if((T->Left=NULL)||(T->Right==NULL)) return T;

else if(Cha(T->Left,p)!=NULL)

return Cha(T->Left,p);

else return Cha(T->Right,p);

}

**c.** **int Muc(Nut \*T, Nut \*p)**

{ if(T==NULL) return 0;

else if(T==p) return 1;

else return 1+Muc(T,Cha(T,p));

}

**Câu 6 a.** **void BoSungCB(CB \*&T, St20 Name)**

{if (T == NULL)

{ T = new CB; T->Ten = Name; T->DSNN = NULL;

T->Left = NULL; T->Right = NULL;

}

else if (strcmp(T->Ten, Name) == 0)

cout << "Ten can bo nay da co!";

else if (strcmp(T->Ten, Name) > 0)

BoSungCB(T->Left, Name);

else BoSungCB(T->Right, Name);

}

**b.** **void InDS(T: TroCB)**

{ if (T == NULL) cout << "Cay rong";

else

{ int i = 1; cout << "TRINH DO…" << endl; DuyetGiua(T, i);}

}

**void DuyetGiua(T : TroCB, int &i)**

{ if (T != NULL)

{ DuyetGiua(T->Left, i); cout << i << "." << T->Name << ":";

TroNN q = T->DSNN;

if (q = NULL) cout << "Khong biet thu tieng nao" << endl;

else

{ cout << q->NgoaiNgu << "-" << q->CapDo; q = q->Next;

while (q != NULL)

{ cout << ":" << q->NgoaiNgu << "-" << q->CapDo; q = q->Next; }

cout << endl;

} i++;

} DuyetGiua(T->Right, i);

}

**Câu 7 a**. **int Min(Nut \*T)**

{ if (T->Left == NULL) return T->Info;

else return Min(T->Left);

}

int Max(Nut \*T)

{ if (T->Right == NULL) return T->Info;

else return Max(T->Right); }

**b. int LaCayNPTK(Nut \*T)**

{ if (T == NULL || (T->Left == NULL&&T->Right == NULL))

return 1;

else if (T->Left == NULL)

return (T->Info <= Min(T->Right) && LaCayNPTK(T->Right));

else if (T->Right == NULL)

return (T->Info > Max(T->Left)) && LaCayNPTK(T->Left);

else return (T->Info <= Min(T->Right)) && (T->Info > Max(T->Left))

&& LaCayNPTK(T->Left) && LaCayNPTK(T->Right);

}

**Câu 8 a.** **int Min(Nut \*T)**

{ int Tam = T->Info; Duyet(T, Tam); return Tam; }

**void Duyet(Nut \*T, int &Tam)**

{ if (T != NULL)

{ if (T->Info < Tam)

Tam = T->Info;Duyet(T->Left, Tam);Duyet(T->Right, Tam);

}

}

**b. int Cap(Nut \*T)**

{ if (T == NULL || (T->Left == NULL && T->Right == NULL))

return 0;

else if (T->Left != NULL && T->Right != NULL) return 2;

else return Max3so(T, Cap(T->Left), Cap(T->Right));

}

**Câu 9 a. void TaoDS(NutCay \*T, NutDS \*&F)**

{ if (T == NULL) F = NULL;

else

{ NutDS \*P; P = new NutDS; P->Info = T->Info;

P->Next = F; F = P;

TaoDS(T->Left, F); TaoDS(T->Right, F);

}

}

**b.** **void DieuChinh(NutCay \*T)**

{ NutDS \*F; TaoDS(T, F); SapXep(F); ChuyenGT(T, F); }

**void ChuyenGT(NutCay \*T, NutDS \*F)**

{ if (T != NULL)

{ ChuyenGT(T->Left, F);T->Info = F->Info; F = F->Next;

ChuyenGT(T->Right, F);

}

}

**Câu 10 a.** **void DienMuc(Nut \*T)**

{ GanMuc(T, 1);}

**void GanMuc(Nut \*T, int Level)**

{ if (T != NULL)

{ T->Muc = Level;

GanMuc(T->Left, Level + 1);

GanMuc(T->Right, Level + 1);

}

}

**b. int DemSoNut(Nut \*T, int Level)**

{TinhMuc(T);int Dem = 0; Duyet(T, Dem); return Dem;}

**void Duyet(Nut \*T, int &Dem)**

{ if (T != NULL)

{ if (T->Muc == Level) Dem++;

Duyet(T->Left, Dem);

Duyet(T->Right, Dem);

}

}

**Câu 1 a void SXNoiBot(a,n)**

{ for(int i=0;i<n;i++)

{ for(int j=n;j>=i+1;j--)

if(a[j]<a[j-1])

{int tam=a[j]; a[j]=a[j-1]; a[j-1]=tam;}

for(int k=0;k<n;k++) cout<<a[k]<<" ";

cout<<endl;

}

}

**Câu 2a void SXChen(int a[],int n)**

{ for(int i=0;i<n;i++)

{ int key=a[i]; int j=i-1;

while(j>=0&&a[j]>key)

{a[j+1]=a[j]; j=j-1;}

a[j+1]=key;

for(int k=0;k<n;k++) cout<<a[k]<<"";

cout<<endl;

}

}

**Câu 3a void SXChon(int a[],int n)**

{ for(i=0;i<n;i++)

{int key=i;

for(int j=i+1;j<n;j++)

if(a[key]>a[j]) key=j;

if(a[key]<a[i])

{ int tam=a[i];a[i]=a[key];a[key]=tam;}

for(int k=0;k<n;k++) cout<<a[k]<<"";

cout<<endl;

}

}

**Câu 4a void SXNhanh(int a[],int Be,int Lon)**

{ if(Be<Lon)

{ int i=Be; int j=Lon; int X=a[(Be+Lon)/2];

while(i<=j){

while(a[i]<X) i++; while(a[j]>X) j--;

if(i<=j){

int tam=a[i];a[i]=a[j];a[j]=tam; i++;j=--;}

}

if(Be<j) SXNhanh(a,Be,j); if(Lon>i) SXNhanh(a,i,Lon);

}

return;

}

**Câu 5 int TKTuanTuLap(int a[], int n, int X)**

{ int i=0;

while(i<n&&a[i]!=X)

i=i+1;

if(i<n) return i; else return -1;

}

**int TKTuanTuDQ(int a[],int i, int n, int X)**

{ if(i>n) return -1;

else if(X==a[i]) return i;

else return TKTuanTuDQ(a,i++,n,X);

}

**Câu 6 int TKNPDeQuy(int x, int a[], int L, int R)**

{ if(L<=R)

{ int m = (L+R)/2;

if(x==a[m]) return m;

else if(x>a[m]) return TKNPDeQuy(x,a,m+1,R);

else return TKNPDeQuy(x,a,L,m-1);

} else return -1;

}

**int TKNPLap(int a[],int x, int n)**

{ int i=0, j=n-1;

do {

int key = (i+j)/2;

if(x==a[key]) return key;

if(x<a[key]) j=m-1; else i=m+1;

}while(i<=j);

return -1;

}

**Câu 7a void ThongKeTanSuat(int a[], int n)**

{ SXNhanh(a,0,n-1); int i = 0;int j=0;

while(i<n)

{ dem =1, j=i;

while(j<n)

{ if(a[i]==a[j+1]) {j++;dem++;}

else break;

}

cout<<a[i]<<dem; i=i+dem;

}

}

**Câu 8a void DieuChinh(int\* a, int i, int n)**

{ int L = 2\*i;int R = 2\*i + 1; int max = i;

if (L <= n && a[L] > a[i]) max = L;

if (R <= n && a[R] > a[max]) max = R;

if (max != i){

swap (a[i], a[max]);

DieuChinh(a, max, n);

}

}

void BuildHeap(int a[], int n){

for (int i = n/2-1; i >= 0; i--){

DieuChinh(a, i, n);

}

}

void Heap(int a[], int n){

BuildHeap (a, n);

for (int i = n-1; i >=0; i--){

swap (a[1], a[i]);

DieuChinh(a, 1, i-1);

}return;

}

**Câu 9 void SXChonLap(int a[],int n)**

{ for(i=0;i<n;i++)

{ int key=i;

for(int j=i+1;j<n;j++)

if(a[key]>a[j]) key=j;

if(a[key]<a[i])

{ int tam=a[i]; a[i]=a[key]; a[key]=tam; }

for(int k=0;k<n;k++) cout<<a[k]<<"";

cout<<endl;

}

}

**void SXChonDQ(int a[], int L, int R)**

{ if(L>=R) return;

else{ int key = L;

for(int i = L+1; i<=R, i++)

if(a[key]<a[i]) key = i;

if(a[L]>a[key])

{int tam=a[i];a[i]=a[key];a[key]=tam;

SapChonDQ(a,L+1,R);}

}

}

**Câu 10 a void SXNhanh(int a[],int Be,int Lon)**

{ if(Be<Lon)

{ int i=Be; int j=Lon; int X=a[(Be+Lon)/2];

while(i<=j){

while(a[i]<X) i++; while(a[j]>X) j--;

if(i<=j){ int tam=a[i];a[i]=a[j];a[j]=tam; i++;j=--;}

}

if(Be<j) SXNhanh(a,Be,j); if(Lon>i) SXNhanh(a,i,Lon);

}

return;

}