**PHÂN TÍCH THIẾT KẾ VÀ THUẬT TOÁN**

**CHƯƠNG I**

**CÂU 1: Định nghĩa độ phức tạp của thuật toán ? So sánh các hàm tính độ phức tạp của thuật toán.**

-Hàm hằng: O(1)

-Hàm logarit: O (logn)

-Hàm tuyến tính: O(n)

-Hàm logarit tuyến tính: O(nlogn)

-Hàm đa thức: O(n^c )

-Hàm mũ: O(C^n)

-Hàm giai thừa: O(n!)

VD:

1. Phân loại các hàm độ phức tạp sau theo 4 nhóm : Hàm hằng, Hàm tuyến tính, Hàm đa thức, Hàm mũ

2^n , (3/2)n, (3/2)^n , 2n^3 , 3n^2 , 1, 1000, 3n

=> hằng: 1, 1000

=> tuyến tính: (3/2)n, 3n

=> đa thức:3n^2, 2n^3

=> hàm mũ:(3/2)^n, 2^n

1. Sắp xếp các hàm độ phức tạp trong Bài tập 1 theo thứ tự độ phức tạp tăng dần.

=> tăng dần: 1, 1000, (3/2)n, 3n, 3n^2, 2n^3, (3/2)^n, 2^n

1. Sắp xếp các hàm sau theo thứ tự độ phức tạp tăng dần:

8n^2 , 6n^3 , 64, nlog6n, log8n, 4n, nlog2n, 8^2n , log2n

=> tăng dần: 64, log8n,log2n,4n,nlog6n,nlog2n,8n^2 , 6n^3,8^2n

**Câu 2: Tính thời gian thực hiện cho các đoạn chương trình không chứa đệ quy (dùng quy tắc cộng, quy tắc nhân).**

\*\* Quy tắc cộng:

Thời gian thực hiện P1 và P2 nối tiếp nhau sẽ là:

T(n) = T1(n) + T2(n) = O(max(f1(n), f2(n))

\*\* Quy tắc nhân:

Thời gian thực hiện P1 và P2 lồng nhau (chẳng hạn vòng lặp lồng nhau) sẽ là:

T(n) = T1(n) × T2(n) = O(f1(n) × f2(n))

VD:

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

int i,j,temp;

for(i= 0; i<=n-2; i++)

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

if (a[j-1] > a[j]) {

temp = a[j-1];

a[j-1]= a[j];

a[j] = temp;

}

}

int tim\_kiem (int x, int a[ ], int n)) {

int found, i;

found = 0;

i = 0;

while (i < n && !found)

if (a[i] == x) 

found = 1;

else

i = i + 1;

return i;

}

**Câu 3: Thành lập phương trình đệ quy và giải bằng phương pháp truy hồi hoặc phương pháp lời giải tổng quát.**

 \*\* tháp hà nội 

void ThapHN(int n, char A, char B, char C) {

if(n > 1) {

ThapHN(n-1,A, C, B);

printf("Chuyen tu %c sang %c\n", A, C);

ThapHN(n-1,B, A, C);

}

}

C1 nếu n=1

T(n)={

2T(n-1) + C2 nếu n>1

\*\*nhị phân

int binarysearch(int x, int \*a, int left, int right) {

if(left > right) return -1;

int mid = (left + right)/2;

if(x == a[mid])

return mid;

if (x < a[mid])

return binarysearch(x,a,left,mid-1);

else

return binarysearch(x,a,mid+1,right);

}

C1 nếu n=1

T(n)= {

T(n/2)+1 nếu n>1

**Chương II**

**Câu 1: Trình bày ý tưởng và minh họa các bước sắp xếp dãy số bằng phương pháp sắp xếp đơn giản: chọn, xen, nổi bọt.**

\*\*Ý tưởng sắp xếp chọn.

Chọn phần tử nhỏ nhất trong n phần tử ban đầu, đưa phần tử này về vị trí đúng là vị trí đầu tiên của mảng hiện hành. Sau đó không quan tâm đến nó nữa, xem mảng hiện hành chỉ còn n-1 phần tử của mảng ban đầu. Bắt đầu từ vị trí thứ 2, lặp lại quá trình trên cho mảng hiện hành đến khi chỉ còn 1 phần tử. Do mảng ban đầu có n phần tử, vậy tóm tắt ý tưởng thuật toán là thực hiện n-1 lượt việc đưa phần tử nhỏ nhất trong mảng hiện hành về vị trí đúng ở đầu mảng.

\*\*Ý tưởng sắp xếp xen

Bắt chước cách sắp xếp quân bài của những người chơi bài. Muốn sắp một bộ bài theo trật tự, người chơi bài rút lần lượt từ quân bài thứ 2, so với các quân đứng trước nó để chèn vào vị trí thích hợp.

\*\*Ý tưởng sắp xếp nổi bọt

Xuất phát từ cuối (hoặc đầu) mảng, đổi chổ các cặp phần tử kế cận để đưa phần tử nhỏ (lớn) hơn trong cặp phần tử đó về vị trí đúng đầu (cuối) mảng hiện hành. Sau đó sẽ không xét đến nó ở vị trí tiếp theo, do vậy ở lần xử lý thứ i sẽ có vị trí đầu mảng là i. Lặp lại xử lý trên cho đến khi không còn cặp phần tử nào để xét.

**Câu 2:Trình bày cách chọn phần tử khóa (pivot), cách phân hoạch và các bước sắp xếp dãy số bằng phương pháp sắp xếp nhanh (Quicksort).**

\*\*cách chọn phần tử khóa (pivot): chọn giá trị khóa lớn nhất trong 2 phần tử có khóa khác nhau đầu tiên kể từ trái qua.

**Chú ý:** Nếu mảng chỉ gồm 1 phần tử hay gồm nhiều phần tử có khóa bằng nhau thì không có chốt.

Ðể phân hoạch mảng: dùng 2 "con nháy" L và R, trong đó L đi từ bên trái và R đi từ bên phải.

• Cho L chạy sang phải tới khi gặp phần tử có khóa ≥ chốt

• Cho R chạy sang trái tới khi gặp phần tử có khóa < chốt

• Tại chỗ dừng của L và R: nếu L < R thì hoán vị a[L], a[R].

• Lặp lại quá trình dịch sang phải, sang trái của 2 "con nháy" L và R cho đến khi L > R.

• Khi đó L sẽ là điểm phân hoạch, cụ thể là a[L] là phần tử đầu tiên của mảng con “bên phải”.

L<R: swap

Chốt v = phần tử > trong 2 phần tử ≠ đầu tiên

{

1. R: stop -> Partition L

**Câu 3: Trình bày quy tắc Pushdown và minh họa cách sắp xếp dãy số bằng phương pháp sắp xếp vun đống (Heapsort) dạng mảng.**

3 cái quy tắc để xoay

**Chương III**

**Câu 1: Trình bày ý tưởng, nhận xét ưu khuyết điểm của các thuật toán: vét cạn, chia để trị, tham ăn, nhánh cận, quy hoạch động, quay lui, cắt tỉa alpha-beta. Nêu ý tưởng áp dụng vào một vấn đề cụ thể.**

**\*\*ý tưởng véc cạn**

Xét tất cả các chu trình, tính tổng độ dài các cạnh của nó, rồi chọn chu trình nhỏ nhất.

**\*\*Ý tưởng quy hoạch động**

Trong thuật toán đệ quy, một số bài toán con phải giải nhiều lần

\*\* tham ăn

Để (X1 + X2 + X3 + X4) nhỏ nhất thì các tờ giấy bạc mệnh giá lớn phải được chọn nhiều nhất.

**\*\*kỹ thuật nhánh cận**

- Ưu điểm: Cho phép tìm được phương án tối ưu

- Khuyết điểm: Cần phải có cách ước lượng giá trị cận tốt mới có thể cắt được nhiều nhánh hạn chế không gian tìm kiếm nhằm nhanh chóng xác định phương án tối ưu.

Mỗi kỹ thuật chỉ phù hợp với một hoặc một số loại bài toán

\*\*\* Mỗi kỹ thuật đều có ưu và khuyết điểm, không có kỹ thuật nào là dùng được trong mọi trường hợp (“trị bá bệnh”)

- Kỹ thuật **Chia để trị** là kỹ thuật cơ bản nhất. Hãy chia nhỏ bài toán để giải quyết!

- Kỹ thuật **Vét cạn** có độ phức tạp quá lớn (lũy thừa)

- Kỹ thuật **“Tham ăn”** giúp nhanh chóng xây dựng được một phương án, dù chưa hẳn tối ưu nhưng chấp nhận được.

- Kỹ thuật **Nhánh cận** cho phép tìm được phương án tối ưu, tuy nhiên cần phải có cách ước lượng cận tốt mới cắt được nhiều nhánh.

- Kỹ thuật **Quy hoạch động** có thể giải được rất nhiều bài toán, tuy nhiên cần phải xây dựng được công thức truy hồi khi số lượng bài toán con phải giải là đa thức

**\*\*Quy hoạch động**

• Ưu điểm: – Chương trình thực hiện nhanh.

– Kỹ thuật quy hoạch động có thể vận dụng để giải các bài toán tối ưu, các bài toán có công thức truy hồi.

• Nhược điểm: Quy hoạch động không hiệu quả khi:

– Không tìm được công thức truy hồi.

– Số lượng bài toán con cần giải quyết và lưu giữ kết quả là rất lớn.

– Sự kết hợp lời giải của các bài toán con chưa chắc cho lời giải của bài toán ban đầu.

**Câu 2: Giải bài toán cái ba lô bằng các kỹ thuật: tham ăn, nhánh cận.**

**\*\*\*\*Cái ba lô bằng thuật toán tham ăn**

Ví dụ 1: Có một ba lô có trọng lượng là 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng được cho trong bảng bên dưới:

|  |  |  |
| --- | --- | --- |
| Loại đồ vật | Trọng lượng | Giá trị |
| A | 15 | 30 |
| B | 10 | 25 |
| C | 2 | 2 |
| D | 4 | 6 |

2.Cho bài toán cái ba lô với trọng lượng của ba lô W = 30 và 5 loại đồ vật được cho trong bảng bên dưới. Tất cả các đồ vật đều chỉ có 1 cái. Giải bài toán bằng kỹ thuật tham ăn (Greedy) ?

|  |  |  |
| --- | --- | --- |
| Loại đồ vật | Trọng lượng | Giá trị |
| A | 15 | 30 |
| B | 10 | 25 |
| C | 2 | 2 |
| D | 4 | 6 |
| E | 8 | 24 |

3. Bài toán mua hàng với số tiền M = 40 và 5 loại hàng hóa được cho trong bảng bên dưới. Giải bài toán bằng kỹ thuật tham ăn (Greedy) ?

|  |  |  |  |
| --- | --- | --- | --- |
| Loại đồ vật | Giá tiền | Giá trị dinh dưỡng | Số lượng |
| A | 10 | 25 | 2 |
| B | 15 | 30 | 1 |
| C | 4 | 6 | 1 |
| D | 2 | 2 | 2 |
| E | 8 | 24 | 1 |

\*\*\*Kỹ thuật nhánh cận: Bài toán cái ba lô

Ví dụ 1: Có một ba lô có trọng lượng là W= 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng được cho trong bảng bên dưới:

|  |  |  |
| --- | --- | --- |
| Đồ vật | TL | GT |
| A | 15 | 30 |
| B | 10 | 25 |
| C | 2 | 2 |
| D | 4 | 6 |

1. Cho bài toán cái ba lô với trọng lượng của ba lô W= 30 và 4 loại đồ vật được cho trong bảng bên dưới. Tất cả các đồ vật có thể được lấy với số lượng không hạn chế. Giải bài toán bằng kỹ thuật nhánh cận ?

|  |  |  |
| --- | --- | --- |
| Loại đồ vật | Trọng lựơng | Giá trị |
| A | 15 | 30 |
| B | 10 | 25 |
| C | 4 | 6 |
| D | 8 | 24 |

**Câu 3: Trình bày quy tắc cắt tỉa alpha-beta và áp dụng định trị nút gốc trên cây trò chơi.**

**Chương IV**

**Câu 1: Định nghĩa và các tính chất của B-cây.**

**ĐN:** B- cây bằng cây tìm tiếm m-phân cân bằng (bậc m).

**tính chất:**

1. Nút góc >= 2 con.
2. Nút trong: [ m/2 ] -> m con.
3. Nút góc -> lá cùng độ dài.
4. Cây tìm kiếm.

**Câu 2: Thực hiện xen, xóa mẩu tin trong tập tin dạng B- cây.**