**20’**

**BBBAB**

**BBADB**

**16’**

**int i=（l+r）/2; r=i ; l=I; K< array[n-1] or r!=n; r; 2 or n(n<2); n ; 3n+1**

**3.**

2 nodes: 2 shapes 3 nodes: 1+2+2=5 4 nodes: 3 can be allocated as 0,3;1,2;2 so 5+5+2+2=14 5 nodes: 14+14+5+5+4=42 **6 nodes: 32+32+14+14+5\*2\*2= 132**

**4.**

BCD

EFGHIJGJ

A

A

B

E

F

CD

GHIJ

A

B

F

C

D

G

I

HJ

A

B

F

C

D

G

I

J

H

E



E

**preorder enumeration: ABCDFEGIHJ**

**5**

**the design logic is to build a Huffman tree**

45%

35%

20%

10%

10%

5%

5%

**D**

**D**

**A**

E

**B**

**C**

**0**

**0**

**0**

**1**

**1**

**1**

**1**

55%

100%

**0**

**Total length: 4 \* 10% +10% \* 3 + 35% \* 2 + 45% = 1.85, the 0-false,1-true as the logic branches.**

**6**

**7.**

**initial: 44, 77, 55, 99, 66, 33, 22, 88, 79**

**pass 1: 44 22 55 33 66 99 77 88 79 pass 2: 44 22 33 55 66 79 77 88 99**

**pass 3: 33 22 44 55 66 77 79 88 99 pass 4: 22 33 44 55 66 77 79 88 99**

**final sorted array: 22 33 44 55 66 77 79 88 99**

**8.**

The task of first phase is to break the files into large initial runs by replacement selection; the second phase is to merge the runs together to form a single sorted run file.

Since working memory is 512KB and the blocksize is 4KB, the working memory holds 128 blocks. The expected runlength is 1024KB, so a single pass of multiway merge forms runs of length 1024KB\*128=128MB. The second pass then forms a run as large as 128MB\*128=16GB.

**9.**

A cluster holds 16\*0.5K = 8K. Thus, the file requires 380/8=47.5clusters.

The time to read a cluster is seek time to the cluster+ latency time + (interleaf factor × rotation time).Average seek time is defined to be 80 ms. Latency time is 0.5 \*8.33, and cluster rotation time is 47.5\*(16/144)\*8.33.Seek time for the total file read time is

47\* (80 + 0.5 \* 600/72+ (16/144)\*600/72)+(80+0.5\*600/72+(8/144\*600/72))=4083.98ms

**10.**  Answer:

H1(22)=0, H1(31)=5, H1(18)=10, H1(35)=6, no conflict

When H1(44)=0, H2(44)=9 （0+9\*1）%11=9，so 44 enters the 9rd slot;

H1(13)=6, H2(13)=2 (6+1\*2)%11=8, so 13 enters the 8th slot;

H1(1)=3, so 1 enters 3 ;

H1(67)=3, H2(67)=10 (3+2\*10)%11= 1 so 67 enters 1(pass by 2)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 |  | 67 | 1 |  | 31 | 35 |  | 13 | 44 | 18 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**11.**

1 to 2: 10 (1,2);

1 to 3: 13(1,2,3);

1 to 4: 12 (1,6,4);

1 to 5: 5 (1,6,5);

1 to 6: 2 (1,6,);

(b)