CSC 3170 Assignment 4 Solutions

Answer Question 1

(a)

$$E(X_N) = \sum_{k=1}^{N} kp_k = \sum_{k=1}^{N} k \times \frac{C}{k} = NC$$

Since the record is given to be present in the file, we have

$$\sum_{k=1}^{N} p_k = 1 \quad \Rightarrow \sum_{k=1}^{N} \frac{C}{k} = 1$$

Or

$$C = \frac{1}{\sum_{k=1}^{N} \frac{1}{k}} = \frac{1}{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N}}$$

Giving

$$E(X_N) = \frac{N}{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N}}$$

(b)

Now,

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N} = 2.93$$

for $N = 10$, giving for the Zipf distribution

$$E(X_N) = \frac{10}{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{10}} = \frac{10}{2.93} = 3.3$$

The corresponding result for the uniform distribution is (10+1)/2 = 5.5. Thus, the number of comparisons is increased by (5.5-3.3)/3.3 = 2.2/3.3 = 67%.

(c) Since the performance of the distribution always outperforms the uniform distribution, the average number of comparison is the same only when $N^* = 1$.

(d) Let R be the record being searched which is not in the file, then when one encounters a record having key greater than the key of R, one would conclude that record R is not in the file and concludes the search – this is the same as looking for the record which is next to R in sequence. Therefore the number of comparison is also approximately $N/2$.

(e)

(i) When the required record is present in the file, this is the same as the sequential search which is $N/2$.

(ii) If the required record is not in the file, then all records have to be compared before one can conclude that the record is not present; in this case the number of comparisons is N.

Answer Question 2

Each node, on the average, will have $n \times 0.69 = 23 \times 0.69 = 15.87$ or approximately 16 pointers and, hence, 15 search key field values. The average fanout is $r = 16$. We can start at the root and see how many values and pointers can exist, on the average, at each subsequent level:

> Level 0 = Root: 16^0 node = 1 node, with 15 key entries, and 16 children pointers
>
> Level 1: 16^1 = 16 nodes, with 16×15 = 240 key entries, and 16^2 = 256 children pointers
>
> Level 2: 16^2 nodes = 256 nodes, with 256×15 = 3,840 key entries, and 16^3 = 4,096 children pointers
>
> Level 3: 16^3 nodes = 4096 nodes, with 4096×15 = 61440 key entries, and 16^4 = 65,536 children pointers
>
> Level 4: 16^4 nodes = 65536 nodes, with 65536×15 = 983,040 key entries, and 16^5 = 1,048,576 children pointers

The number of entries for a tree of height 2 = 3,840 + 240 + 15 = 4,095 entries on the average

The number of entries for a tree of height 3 = 61440+4095 = 65535 entries on the average

The number of entries for a tree of height 4 = 983040+65535 = 1,048,575 entries on the average

Now, let S be the total number of entries that the tree of height h holds. We have for:

Level 0, the number of entries for that level is $r^0 \times (r-1)$

Level 1, the number of entries for that level is $r^1 \times (r-1)$

Level 2, the number of entries for that level is $r^2 \times (r-1)$

Level 3, the number of entries for that level is $r^3 \times (r-1)$

Level h, the number of entries for that level is $r^h \times (r-1)$

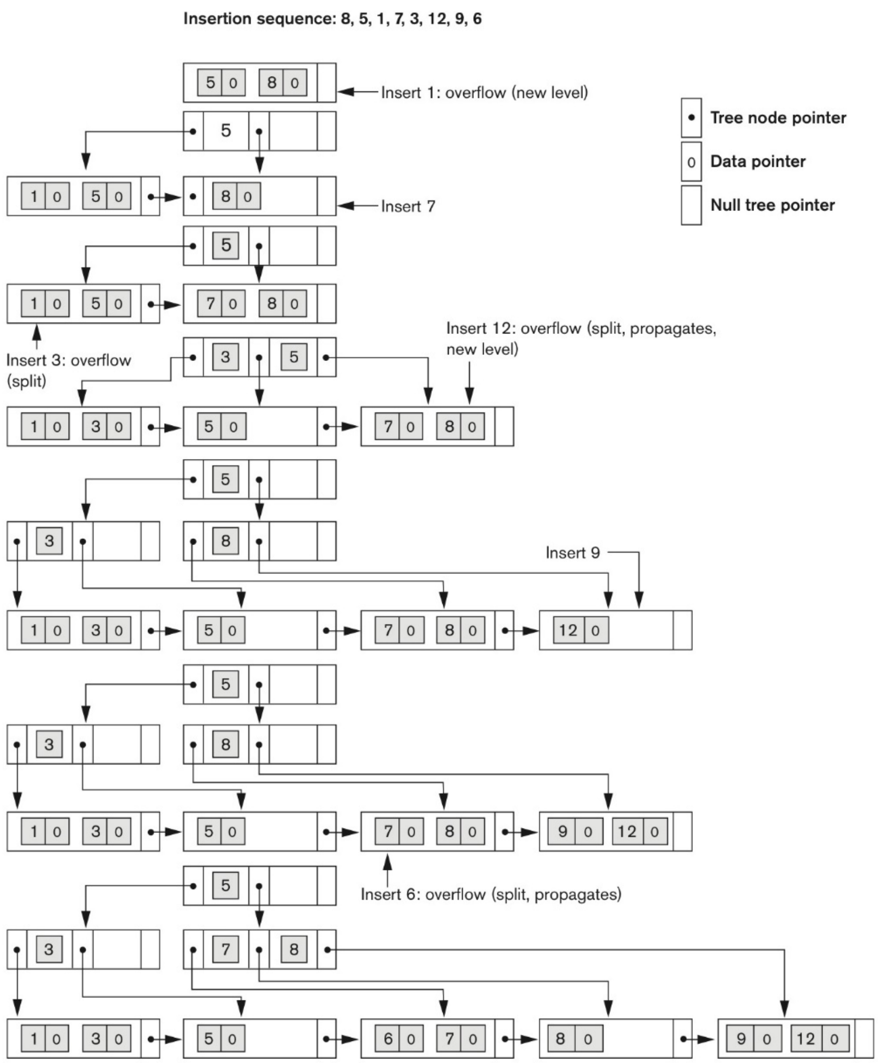
Thus, the total number of entries for a tree of height h is

$(r-1) \times [r^0 + r^1 + r^2 + \ldots + r^h] = (r-1) \times (1 - r^{h+1})/(1-r) = (r^{h+1} - 1)$

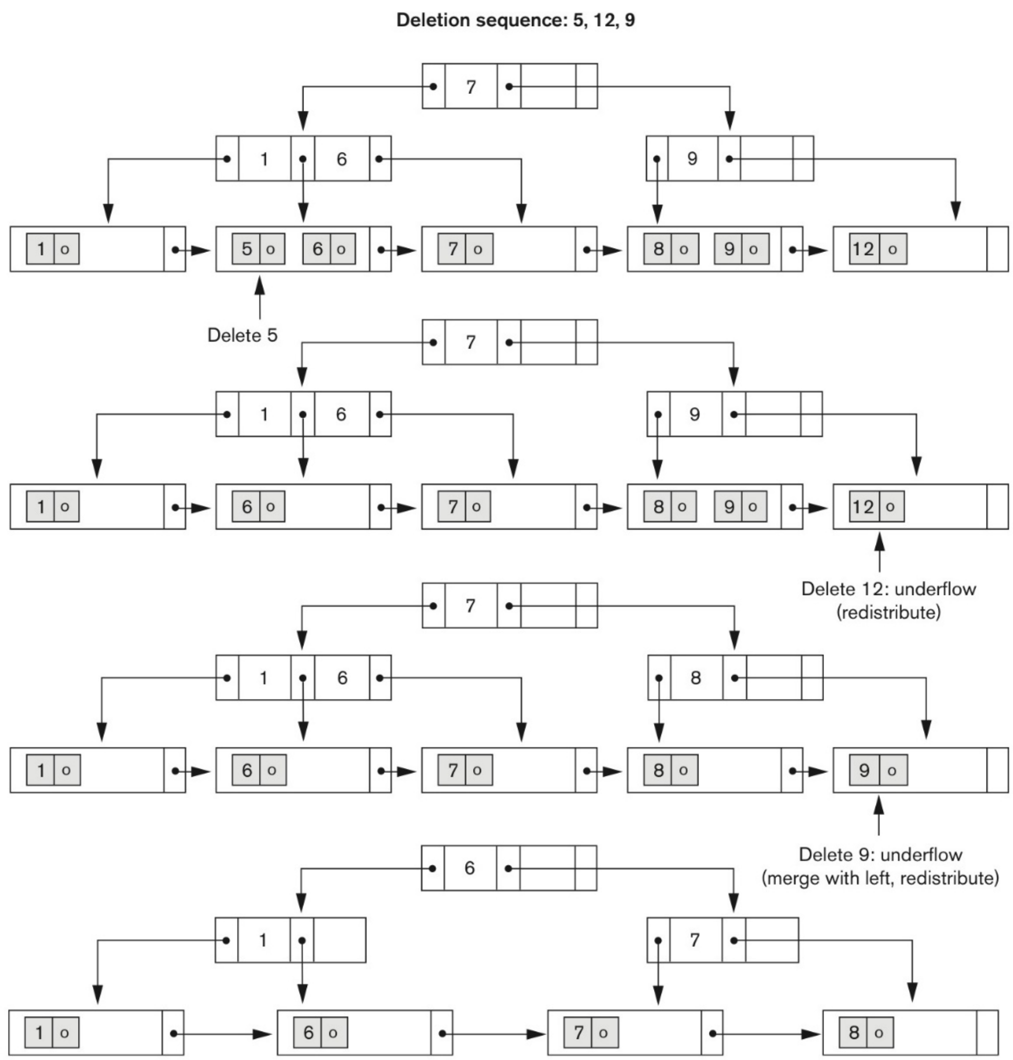
That is, we obtain:

$$S = (r^{h+1} - 1)$$

Answer Question 3



Answer Question 4



Answer Question 5

(i) the mean storage utilization E(ρ) is

$$\frac{f}{f'}\ln\frac{1}{f}$$

which equals 3*ln (4/3) = 3*0.288 = 86.3%

(ii) the variance of the storage utilization Var(ρ) is

$$\sigma_f^2 = f - \left(\frac{f}{f'}\right)^2 \left[\ln\left(\frac{1}{f}\right)\right]^2$$

which equals to ¾ - 9*[ln (4/3)]^2, the square root of which gives sd(ρ) = 0.072.

(iii) The best is to make use of the cumulative distribution function G(.). The cdf is

$$G(x) = \frac{1}{f'}(1 - \frac{f}{x})$$

Therefore Prob [0.8 ≤ ρ ≤ 0.9] = G(0.9) – G (0.8) = 0.67-0.25 = 0.42.

Alternatively, one can use the probability density function g(.) and integrate. Both approaches are acceptable.

$$\text{Prob } [0.8 \le \rho \le 0.9] = \int_{0.8}^{0.9} \frac{f}{f'x^2}\, dx = 3 \times \left[\frac{1}{0.8} - \frac{1}{0.9}\right] = 0.42.$$

(iv) The median *m* can be obtained from the cdf

$$G(x) = \frac{1}{f'}\left(1 - \frac{f}{x}\right)$$

The median *m* is Prob [*f* ≤ ρ ≤ *m*] = 0.5, which gives G(*m*) – G (*f*) = G(*m*) = 0.5. That is, we solve

$$G(m) = 3\left(1 - \frac{3}{4m}\right) = 0.5$$

which gives *m* = 6/7, or 85.7%, which is slightly less than the mean of 86.3%. Therefore, the probability density exhibits a very slight positive skew (i.e. skewed to the right).

(v) From the above, the median m can be obtained from the cdf as

$$G(m) = \frac{1}{f'}\left(1 - \frac{f}{m}\right) = \frac{1}{2} \ .$$

From this, we obtain

$$m = \frac{2f}{1+f} \ .$$

Answer Question 6:

(i) The records will hash to the following buckets:

K $\rightarrow$ h(K) (bucket number)

2305 $\rightarrow$ 1,

1168 $\rightarrow$ 0,

2580 $\rightarrow$ 4,

4871 $\rightarrow$ 7,

5659 $\rightarrow$ 3,

1821 $\rightarrow$ 5,

1074 $\rightarrow$ 2,

7115 $\rightarrow$ 3,

1620 $\rightarrow$ 4,

2428 $\rightarrow$ 4 overflow

3943 $\rightarrow$ 7,

4750 $\rightarrow$ 6,

6975 $\rightarrow$ 7 overflow

4981 $\rightarrow$ 5,

9208 $\rightarrow$ 0,

Two records out of 15 are in overflow, which will require an additional block access. The other records require only one block access. Hence, the average time to retrieve a random record is:

(1 * (13/15)) + (2 * (2/15)) = 0.867 + 0.266 = 1.133 block accesses

(ii)

Record#	K	h(K) Bucket Number	Hash Value
Record 1	2305	1	00001
Record 2	1168	16	10000
Record 3	2580	20	10100
Record 4	4871	7	00111
Record 5	5659	27	11011
Record 6	1821	29	11101