Samsung Interview Experience | Set 8 (On-Campus for Samsung Research Institute, Noida)

Samsung visited our campus.

I. Written test :

Online test consisting of a coding question, which was to be solved in 3 hours.

Given an integer ‘K’ and a tree in string format. We had to print the sum of all elements at Kth level from root.

For example:-

```
                0
            /     \
        5          7
   /    \      /   \
  6       4     1      3
                \
                  9
```

Tree was given in the form: (node value(left subtree)(right subtree))

For tree given above: (0(5(6()())(4()(9()())))(7(1()())(3()())))

Input format: K Tree

Output format: Sum

For example, for given tree:

Input: 2 (0(5(6()())(4()(9()())))(7(1()())(3()())))

Output: 14

Students, who were able to pass all the test cases, shortlisted for next round.

II. Technical Interview : (40 Min.)

1. Tell me about yourself.

2. A discussion on my project.

3. Interviewer asked me whether I know android programming or not.

4. What is polymorphism?

5. Difference between overloading and overriding.

6. Implement queue using linked list.

7. Few basic questions on sorting.

8. Interviewer asked me to tell four header files in C.

9. Full form of ‘conio’(header file).

10. A simple puzzle:

https://en.wikibooks.org/wiki/Puzzles/Logic_puzzles/Pair_of_Socks

11. What is semaphore?

12. He asked about CPU scheduling algorithms.

13. Difference between preemptive and non- preemptive algorithms.

14. Round Robin algorithm.

15. He asked few SQL queries.

16. What is view in DBMS?

17. What is Indexing?

18. Do you have any question?

III. HR Interview : (30 Min.)

1. Tell me about yourself.

2. Discussion about college life.

3. He asked about the best phase of my life.

4. What is your biggest achievement?

5. Do you have offer from any other company at present?

6. Why Samsung?

7. A puzzle:

```
      [ ] [ ]
  [ ] [ ] [ ] [ ]
      [ ] [ ]
```

He asked me to fill the numbers from 1 to 8 in these 8 boxes, one number exactly once, such that no two consecutive elements are adjacent(diagonally, horizontally or vertically) to each other.

```
        Solution:       3 5
                      7 1 8 2
                        4 6
```

8. Do you have any question?

Finally, 17 students were selected.

Samsung R&D Interview Experience | Set 37 (For developer profile)

Samsung R&D Bengaluru, visited our campus for full time recruitment. There were 5 rounds in total.

1) Online coding round

2) GD

3) Technical round 1

4) Technical round 2

5) HR

Round 1: Online Coding round

This was a 3 hours coding round in which we had to code 1 problem having 10 test cases. Only those students were selected for the next round who passed all the test cases.
Note- You can compile your code as many number of times as you want, but a maximum of 5 submissions were allowed to test on the given test cases.

Here is the question:-

Mr. Kim has to deliver refrigerators to N customers. From the office, he is going to visit all the customers and then return to his home. Each location of the office, his home, and the customers is given in the form of integer coordinates (x,y) (-1<x<101, -1<y<101) . The distance between two arbitrary locations (x1, y1) and (x2, y2) is computed by |x1-x2| + |y1-y2|, where |x| denotes the absolute value of x; for instance, |3|=|-3|=3. The locations of the office, his home, and the customers are all distinct. You should plan an optimal way to visit all the N customers and return to his among all the possibilities.

You are given the locations of the office, Mr. Kim's home, and the customers; the number of the customers is in the range of 5 to 10. Write a program that, starting at the office, finds a (the) shortest path visiting all the customers and returning to his home. Your program only have to report the distance of a (the) shortest path.

You don't have to solve this problem efficiently. You could find an answer by looking up all the possible ways. If you can look up all the possibilities well, you will get a perfect score.

[Constraints]

4<N<11. Each location (x,y) is in a bounded grid, -1<x<101, -1<y<101, and x, y are integers.

[Input]

You are given 10 test cases. Each test case consists of two lines; the first line has N, the number of the customers, and the following line enumerates the locations of the office, Mr. Kim's home, and the customers in sequence. Each location consists of the coordinates (x,y), which is reprensented by 'x y'.

[Output]

Output the 10 answers in 10 lines. Each line outputs the distance of a (the) shortest path. Each line looks like '#x answer' where x is the index of a test case. '#x' and 'answer' are separated by a space.

[I/O Example]

Input (20 lines in total. In the first test case, the locations of the office and the home are (0, 0) and (100, 100) respectively, and the locations of the customers are (70, 40), (30, 10), (10, 5), (90, 70), (50, 20).)

5 (Starting test case #1)

0 0 100 100 70 40 30 10 10 5 90 70 50 20

6 (Starting test case #2)

88 81 85 80 19 22 31 15 27 29 30 10 20 26 5 14

Output (10 lines in total)

#1 200

#2 304

HINT:- Use BackTracking

2) GD

We were given a problem in ML.

Problem was to predict the no. of cycles would be required by a cycle renting company(like OLA) at a given point of time. Given previous data with various features of service like time of service, date , city name, humidity , temp , etc predict no of cycles required in that city at give time.

We discussed about linear regression , feature reduction, naive bayes prediction , classifiers , feature selection techniques.

I thought the idea was to keep solution as simple as possible initially . For such task linear regression would be good , Jumping to ANN would not help in front of them.

We also discussed about how to store this data on disk or hdfs etc.

For storing data , we discussed about storing in SQL or Nosql databases. Advantages and disadvantages. How we will create indexes for fast access. It lasted for 40 mins.

6 were shortlisted out of 9.

3) Technical round -1

1. Introduce yourself
2. Area of interest.
3. Linear regression, write code for gradient descent, Stochastic gradient descent, feature reduction and selection techniques, information gain , decision trees, random forest .
4. Then he asked question about my research posters from my CV.
5. He asked me to create a question answering system . where we discussed about document indexing , merging posting lists , pos tagging , named entity recognition ,stemming ,stop words removal etc (Basically he was testing me in NLP , IR , ML), we discussed this for 20-25 minutes . Interviewer was very helpful.
6. After this, we discussed about deep learning approaches for question answering system.
7. Questions on tensorflow , Numpy , diff between python and C++ .

It lasted for about 1 hr.

4) Technical round -2

1. Various classifiers like SVM (in depth) , decision trees vs random forest , Bagging boosting , etc.
2. Discussion about my internship project on image classification.
3. Create a system to convert speech to handwritten documents. We discussed about how to recognise one’s handwriting , language modelling , spelling correction , feedback from user, etc
4. What new can I see in mobile security using ML ?
5. Real life applications of all my projects written in CV.
6. And then final discussion about my favourite project among those.

It lasted for about 1 hr.

5) HR round

In both technical interviews they asked me from my CV . so Defend your CV well.
From interviews I learned , Given a problem one should approach step wise to bring solution . Like given a ML problem , one should not straight away jump to ANN and RNN , etc . Initially give them a simple solution of a problem (like a brute force) and then build up your discussion with further improving it. It will reflect your depth in subjects to interviewer.

If you are stuck something ask interviewer for help .

Best of luck

Samsung Interview Experience | Set 11 (On-Campus for Samsung R&D)

Written round: One programming question of 3 hours duration.

There were two rounds : tech PI and HR round.
A. Tech. PI : Firstly,interviewer went through my whole resume.He started asking questions from project done during summer internship.Questions were:
1. What is he project about?
2. Difference between 3G and 4G.
3. What is the aim of this research internship?
4. How much was i able to complete in two months?(I told that i am continuing as major project,since in two months time i could not get desired result).
Then, he went onto programming language i was comfortable. I answered C.
1.sizeof(“shubham”)
strlen(“shubham”)
I was asked to write output.Output would be 8 and 7. He asked why. Basic concepts should be clear.
2.Storage Classes in C-asked me concept and gave some questions to write

output.
3.Constant pointers and pointers to constant. Asked to write output of snippets he gave.
4.malloc,caloc,realloc,free-syntax
5.Questions on pointers.

Basic concepts of let us C were asked mostly,that book is enough . Also see, GEEKSFORGEEKS, C section.

Operating System: Mutex and Semaphore,critical Section, Scheduling algorithms, Producer-Consumer Problem.
(Data Structures was not asked to me but others.)

Puzzles:
1. Maximum run a batsman can make in 50 overs.
2. There are 10 coins. 5 face head side and 5 as tail. You are blind folded.Make two piles of equal number of coins such that number of head in equal in both the piles.
Assumption:You can flip coin any number of times.
Question: Why do you want to join Samsung,though placed in OFSS?

Questions: Any plans of future studies. I answered ‘NO’ only job.

Last question: Any question I would like to ask them. I asked what work would be doing in R&D. He explained me a lot of things.

B. HR Round:
1. Introduce yourself.
2. How would your friends describe you?
3. Tell the recent experiences of which makes you initiative.(i told that i am initiative,then he asked that question. So be ready with the examples,when u say any of your qualities.)
4. What is your weakness?
5. Puzzles.He asked me to fill the numbers from 1 to 8 in these 8 boxes, one

number exactly once, such that no two consecutive elements are adjacent(diagonally, horizontally or vertically) to each other.
https://www.geeksforgeeks.org/samsung-rd-sri-noida-interview-experience-set-8-on-campus/
6.Which places in India have you been and what is there to see in Raipur ?
If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share

Samsung Interview Experience | Set 36 (On-Campus)

Samsung Bangalore came to our campus to recruit for a software development engineer role. The following process took place.

Round 1: Online test.
The question was to calculate the shortest time taken from source to destination in outer space. We had n bidirectional warm holes in between which could be used for faster travel. STL or any other extra libraries were not allowed. There were 50 test cases and all had to be passed in order to clear this round.
A total of 22 students got shortlisted for the next round.
Round 2: Technical group discussion.
We were divided into two groups and were given a question and had to come up with approaches and discuss them with the group. Everyone was even made to compile their thoughts on a piece of paper. Separate questions were asked from each of the groups.
Q1. Given large number of points on a 2D plane come up with a fast approach to calculate n closest point to a given x,y point on the plane.
Q2. Given two files having unique characters and about 100 gb in size. One has some characters missing. You need to compare with the original file and find

the count of missing characters.

I did really well in my GD. I got the second question and came up with a binary search approach which impressed the panel. I later suggested some other optimizations.

Finally, 12 people got shortlisted. You were not expected to solve the whole question. Any good approach or active involvement was enough to get shortlisted.

Round 3: Technical Interview 1

I was the first one to be called for the interview. The guy asked me to give a brief introduction of myself. After that he straight away jumped to my internship work. We had an elaborate 15 min discussion on the work I did at a start up. Then he started asking me some basic OS stuff. I was asked to code the producer consumer problem. I did that correctly. After that he started diving deep into OS and memory concepts, of which I had little clue. I did poorly there. After that we had a 5 min discussion on one of my projects which was on neural networks. At last he asked me if I had any questions for him I said I would like to know more about Samsung and the research areas that they are presently working on. He gave a pretty informative reply.

Round 4: Technical Interview 2

This was a pure algorithmic problem solving round. The interviewer without wasting anytime gave me graph question to solve.

A truck has to go from one city to another. There are n cities in between connected by roads and the time taken to reach cities are given as edge weights. We had to find the shortest path given that truck driver can only drive for 12 hours in a day and has to rest in a city after that. My initial approach was to do it by dfs since we had to only find the shortest number of days. To which he said he wanted both days and hours.Finally, I came up with a modified Dijkstra’s approach. He seemed satisfied.

I was asked another dp question where we had n wines and had to sell from either left or right making maximum profit in the process. We could sell once

each year and the cost of the wines keeps getting increased with each passing year. I quickly came up with a solution for this problem.
The last question was same as the GD topic for the other group. After a lot of thinking and clarification I came up with an approach which requires dividing plane into grids and saving points in each grid and calculating for only those grids which were closest to the given point. I used max heap for this. Didn’t code anything.

Round 5: HR

This was more strenuous than expected.
He grilled me on several questions like-
Why is your CGPA low ? It was 7.
You have a major in electronics then why IT ?
Proudest moment of your life ?
Any plans of doing an MBA or MS ?
The round ended with me asking him about my role at Samsung if I get selected.
The final list was released after an hour. They took four students. I was one of them and the only non cs candidate.
Tips: Know everything you have mentioned in the resume. If you have mentioned anything regarding ML, they almost certainly will grill you on that.
Solve previous year interview questions.
Samsung Interview Experience | Set 10 (On-Campus for Samsung Research Institute, Noida)

I. Written test : Same question asked in Set 8.(lucky for us)3 hrs were given to solve the question.
Just keep in mind that dont submit the solution too early.They maintain record of all the submission time
and they will ask that how you were able to solve the question so quickly.(They asked me)

They had installed their own software.Only visual studio,eclipse,notepad and calculator was allowed.
If you tried to open any other tab it was considered as cheating.So keep care.Interface not so friendly.
And only three submissions were allowed so if you are 100% sure that your code will pass all the test cases
then only submit.
Students, who were able to pass all the test cases, shortlisted for next round.
A total of 41 students were selected.

Results were announced around 3PM. I got call from my TPC that i have to give my
interview within 5 mins.I was like “really itni jaldi???”

Technical Round:
This was longest interview i have given.Lasted for 2.30 hrs.
The moment i entered the cabin he asked me “Did you eat your lunch”??
I was confused at that moment but later i realized that why did he asked..
Asked me to introduce myself...about my family…
Then asked which language you prefer C/C++ or JAVA
I said “C/C++ is like Hindi for me and JAVA is like English”
He smiled.Then he asked which papers i had studied and asked me to rate myself.
Then he started to ask question.

From C/C++:

1. Storage Classes in C-asked me almost every thing.(long discussion)
2. Memory layout of C programs.(he was looking for the details)
3. structure and union (every minute detail)
4. Complicated C declaration (asked to write prototype of functions)
5. Constant pointers and pointers to constant(he asked me to explain through

diagram)

6. Structure padding and alignment.(why padding,how does it helps,how do we force complier not to allow padding)

7. Dangling pointers,malloc,caloc,realloc,free

Quite long discussion.Other questions like how does free operator knows how much memory

to freed.A lot more questions.

8. Questions on pointers(single pointer,double pointer and a lot)

9. About MACROS and preprocessor.

10. Preincrement and postincrement operator(was asked to write output and logic behind it)

11. Inline functions,dyanamic arrays,enum,

and a lot

12. about OOPS

Plz go through C section on Geeksforgeeks…U will not miss any of these.

From OS:

1. Scheduling algorithms(long discussion)

2. Mutex and seamphores(in detail)

3. Multiprogramming,Multitasking,Multiprogramming etc.

4. Deadlock(asked to explain deadlock to layman,how will you create a deadlock and a lot)

5. Schedulers(their working)

6. states of programms-like ready state,waiting state…was asked to draw digram

7. Producer-Consumer Problem

Coding:

1.Implement Stack using link list(only condition was that return type of push and pop should be void,solved by using pointer to pointer)

2.Implement you own strcat() function.Dont use string header.

3.Swap two numbers without using arithmetic operator.

Keep care when you write the code.He will go throughly through the code.So take your time and dont be
in a hurry.

After that I was called for HR Round immediately. I was the first one for HR.

1. Asked me to introduce myself.about my family.
2. About my hobbies,interest.
3. A small technical question.
4. Puzzle-i dont remeber..didint solved it comletly
5. Then asked why samsung.He wanted specific reasons.

No prainsing answer like samsung has a good brand value etc.

6. Asked me whether i am already placed or not.
7. Then in which companies i was rejected and asked why??
8. Why should we hire you??
9. Any thing you would like to ask??

I asked about typical work day and cuurent samsung project ..and then he explained me a lot of things..
He was very cool and interacted a lot.
HR round lasted for 35-40 minutes.

Thanks to GeeksForGeeks team for developing such a wonderful site.It helped me a lot in the preparation.

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Samsung Interview Experience | Set 30 (On-Campus)

Samsung R&D Noida visited our campus for full time recruitment and internships. There were 4 rounds in total.

1) Online coding round

2) Technical round 1

3) Technical round 2

4) HR

Round 1: Online Coding round

This was a 3 hours coding round in which we had to code 1 problem having 10 test cases. Only those students were selected for the next round who passed all the test cases.

Note- You can compile your code as many number of times as you want, but a maximum of 5 submissions were allowed to test on the given test cases.

Here is the question:-

Mr. Kim has to deliver refrigerators to N customers. From the office, he is going to visit all the customers and then return to his home. Each location of the office, his home, and the customers is given in the form of integer coordinates (x,y) (-1<x<101, -1<y<101) . The distance between two arbitrary locations (x1, y1) and (x2, y2) is computed by |x1-x2| + |y1-y2|, where |x| denotes the absolute value of x; for instance, |3|=|-3|=3. The locations of the office, his home, and the customers are all distinct. You should plan an optimal way to visit all the N customers and return to his among all the possibilities.

You are given the locations of the office, Mr. Kim’s home, and the customers; the number of the customers is in the range of 5 to 10. Write a program that, starting at the office, finds a (the) shortest path visiting all the customers and returning to his home. Your program only have to report the distance of a (the) shortest path.

You don’t have to solve this problem efficiently. You could find an answer by looking up all the possible ways. If you can look up all the possibilities well, you will get a perfect score.

[Constraints]

4<N<11. Each location (x,y) is in a bounded grid, -1<x<101, -1<y<101, and x, y are integers.

[Input]

You are given 10 test cases. Each test case consists of two lines; the first line has N, the number of the customers, and the following line enumerates the locations of the office, Mr. Kim’s home, and the customers in sequence. Each location consists of the coordinates (x,y), which is reprensented by ‘x y’.

[Output]

Output the 10 answers in 10 lines. Each line outputs the distance of a (the) shortest path. Each line looks like ‘#x answer’ where x is the index of a test case. ‘#x’ and ‘answer’ are separated by a space.

[I/O Example]

Input (20 lines in total. In the first test case, the locations of the office and the home are (0, 0) and (100, 100) respectively, and the locations of the customers are (70, 40), (30, 10), (10, 5), (90, 70), (50, 20).)

5 (Starting test case #1)

0 0 100 100 70 40 30 10 10 5 90 70 50 20

6 (Starting test case #2)

88 81 85 80 19 22 31 15 27 29 30 10 20 26 5 14

Output (10 lines in total)

#1 200

#2 304

HINT:- Use BackTracking

Tips:- Practice questions on DFS, BFS, Backtracking

Round-2 (Technical Interview Round-1)

The interviewer saw my resume. He was quite impressed by my resume.

1) Discussion on my projects.

2) Questions related to OS:-

a) What is Demand Paging

b) Segmentation

c) Virtual Memory

d) Segment Table

e) Offset and base address

f) Shallow copy and deep copy

g) Semaphore

h) Swap In and Swap Out

i) Starvation and Aging

3) Questions on Data Structure

a) Delete node from bst

b) worst case scenario in bst

4) A Simple Puzzle

Round-3 (Technical Interview Round-2)

1) Tell me about yourself

2) Rate yourself in OS, C, C++ (out of 10)

3) Delete key from linked list

4) Dynamically allocate 2d array

5) Memory Layout

6) Write a program to find when we get stack overflow if we are using recursive functions.

6) Again Discuusion on My projects

And then he asked me if I had any question.

Round-4 (HR)

1) Tell me about yourself which is not mentioned in your resume.

2) Why samsung?

3) What is your criteria to select a job?

4) Why should we hire you?

Overall, It was nice experience . Finally I got selected 🙂 (Total 20 were selected)

Samsung R & D Noida Question September 2018

Round 1: There was an online coding round on 10th of September, 2018, For which about 50 candidates had come. I will come directly to that special question, for which one has to solve in 4 hours, and there were 50 test cases to be passed for that problem. Any one passing all test cases would be considered for next round of interview.

The question was like this…

You have to place an electronic banner of a company as high as it can be, so that whole the city can view the banner standing on top of TWO PILLERS.

The height of two pillers are to be chosen from given array.. say [1, 2, 3, 4, 6].

We have to maximise the height of the two pillars standing side by side, so that the pillars are of EQUAL HEIGHT and banner can be placed on top of it.

In the above array, (1, 2, 3, 4, 6) we can choose pillars like this, say two pillars as p1 and p2..

Then pillars can be,

p1 = 3 unit… Choosing element (3) from array,

Similarly p2 = 3 choosing (2 + 1) from array.

Since, two pillars are equal, we can put board on it…

But we have two maximise the height of the pillars,

And if we check for other heights, we can see p1 = 6 p2 = 4 + 2 which is greater than 3 (the previous height)..

We have to see if we can further maximize the height… Yes it can be 8.

I.e. p1 = 6 + 2 = 8. p2 = 4 + 3 + 1 = 8.

Both pillars are equal and banner can be placed… And since this is the maximum height attainable for two pillars, we print the answer as 8. In case, there is no combination possible, print 0 (zero).

INPUT :

```
1
5
1 2 3 4 6
```

First line is T number of test cases to be followed.

Second line of input is number of different pillars.

Third line of input is different available heights of pillars.

Note : heights of given pillars can be same .. I.e. array can have same elements repeated.

Output.

```
8
```

Simply print the maximum height attainable so that board/ banner can be placed.

In case there is no possible combination for placing banner with equal weighted pillars, then print 0.

Constraint's : some general constraints were given but I don't remember exactly.

This was the question to done in 4 hours.

I was out after first round, as I could not solve this problem. Hope this may help you for cracking Samsung interview.

Samsung Interview Experience | Set 13 (Off-Campus)

Round 1

1 coding question (3 hour): m-coloring problem with m=2. You have to run all 10 test cases to qualify.

Round 2: Face to face Interview (30 min) after 2 weeks of written test.

- Software Design patters
- Factory pattern
- Singleton pattern
- Discussion on projects done in college
- Spring framework
- MVC model
- Hibernate

Basically this round was based on Resume.

All above things were mentions in my resume so he asked.

Round 3: Face to face Interview (45 min)

1. Discussion on the coding round question. He asked me to code it again and time complexity.

2. Test if a number is power of 2 or not in one line code.

https://www.geeksforgeeks.org/write-one-line-c-function-to-find-whether-a-no-is-power-of-two/

3. egg droping puzzle

http://quiz.geeksforgeeks.org/puzzle-set-35-2-eggs-and-100-floors/

4. Extended previous puzzle

https://www.geeksforgeeks.org/dynamic-programming-set-11-egg-dropping-puzzle/

5. Implement sizeof operator

https://www.geeksforgeeks.org/implement-your-own-sizeof/

Round 4: HR (20 min)

- Discussion on interviews.
- Why I want to leave current job.
- Why samsung.

Overall It was easy.
You just have to be confident.

Thanks

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Samsung Interview Experience | Set 18 (On-Campus)

Coding round was difficult, recently Samsung has increased difficulty level of its coding round, so please be in touch with DFS, BFS, Graph, matrix and ticker questions of backtracking.

Technical Interviews were easy to crack as compared to coding round. I had 3 rounds of interview

Round 1

Interview went through my resume and asked about the project I did

1 He asked me about BFS and DFS
2 Asked how to find cycle in directed graph? DFS or BFS ? Why not BFS?
3 Topological sorting ?
4 Radius of a graph? in O(n) complexity
5 Asked question about linked list and array.

Round 2

1 Gone through my resume, I had mention internet security as area of interest. He asked my symmetric key and asymmetric key cryptology.
2 Asked pointer to a function and function to pointer.
3 Asked to flip a bit in number.

Round 3

1 Detailed discussion of the project I had mention in my resume, challenges and how I resolved them.
2 Asked about memory layout in c programs.
3 Asked about static, register, auto and other access modifier.

Strong suggestion for you guys, if they are hiring you for project lead position then please better you stay unemployed and reject this offer. Project lead sounds nice but reality is very heart-breaking. You work as educated clerk with data entry work. Now if feel trapped and sad for my last job. All the best guys hope you take my advice before you join Samsung Noida as a project lead. If you like to work on new technology then please don't join Samsung Noida as a project lead (PL).

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Samsung Interview Experience | Set 7 (On-Campus for Samsung Research Institute)

There were total of three rounds:

1. Written Test One question (time: 3Hrs)

The questions were as follow:

A company sells its products with a unique serial number on it. Company has has found that there are some products that don't sell well which are identified to have ominous numbers in the serial number of the product. So if a serial number of the product contains atmost 'k' ominous number then it won't sell.

Given a range form s to e, you need to find number of products that would sell, leaving out the products that contains atmost ’k’ ominous numbers.

Input: First line contains the number of test cases, followed buy the range s to e, 1

2. After clearing this round, I was called for the Technical Interview. The Interviewer asked me to breif myself and the projects I had taken up in my final year, along with questions from geeksforgeeks it self. He asked me to declare a two dimensional array using pointers in C++, he asked me to find the next greater element [link: https://www.geeksforgeeks.org/next-greaterelement/], two DP questions.
1. Find longest increasing sub-sequence in 2d array.
[link: http://stackoverflow.com/questions/6558710/longest-increasingsequence–2d-matrix-recursion],

2. Kandane’s Algorithm. Here the focus was more on the attitude towards the problem solving and the approach. Also they insisted me to get the optimized solution. I was able to give the optimized solution and he was satisfied.

3. After this I was called for the HR round. One question that I remember was, ‘given a pen and the technologies like artificial intelligence, hadoop, and what whatever you can thin of, what are the three new features that you would add to this pen?’ Then he asked me about my interests in Computer Science, highschool-education, how i spend my time in college apart from the academics and few managerial HR questions. You have to remember here, you need to stand for whatever you say, he may create scenarios, in which you will want to change your answers and evaluations, but if you feel you were right then and now again stand for your point. I hope you have got some insight

about the interview process. Good Luck and keep reading geeksforgeeks, because interviewers also pick up questions from the very same site.

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

All Practice Problems for Samsung !

Samsung Interview Experience | Set 20 (R & D Banglore)

Round 1: Online coding round
There is dedicated Samsung software for coding test the question is given below:
There is one spaceship. X and Y co-odinate of source of spaceship and destination spaceship is given. There are N number of warmholes each warmhole has 5 values.
First 2 values are starting co-ordinate of warmhole and after that value no. 3 and 4 represents ending co-ordinate of warmhole and last 5th value is represents cost to pass through this warmhole. Now these warmholes are bi-direction.
Now the to go from (x1,y1) to (x2,y2) is abs(x1-x2)+abs(y1-y2).
The main problem here is to find minimum distance to reach spaceship from source to destination co-ordinate using any number of warm-hole. It is ok if you wont use any warmhole.
Solution:
You can make graph which contain edge between all points and put cost of that edge. Now apply dijstra algorithm to find minimum distace between source to destination co-ordinate of spaceship.
Here the main catch is that the cost to pass through warmhole can be zero so you have to take care while making graph matrix.

After this round 3 are shortlisted out of 60.

Round 2:Technical Round on SkyPe.

-Tell me about yourself.

-I have done project in Android so he asked me the difference between activity and fragment.

-what is the difference between process and thread and some detain question about thread.

-what process scheduler do,type of scheduler and its algorithm.

-find path in matrix with min. cost staring from left upper corner to right bottom corner.

https://www.geeksforgeeks.org/dynamic-programming-set-6-min-cost-path/

-topological sorting of graph

https://www.geeksforgeeks.org/topological-sorting/

-You have 1 TB of big file containing numbers and you have to find to N smallest number.

Round 3:HR Round

-Tell me about yourself.

-Tell me about your final year project.

-why you want ot join samsung.

-strength and weakness.

Samsung Interview Experience | Set 9 (On-Campus for Samsung Research Institute, Noida)

First Round : Coding Round

Second Round : Tech + HR interview

First Round :

it was quite simple, the question was similar to , given a level K , you have to find out the sum of data of all the nodes at level K in a binary tree.

it was bit tricky the input is in the form

(P(C()())(C()()))

P is for Parent, C is for child.

if parent has one child : (P(C()())())

if parent has no child : (P()())

Second Round:

Technical:

1. write a program , given a matrix with 0’s and 1’s , you enter the matrix at (0,0) in left to right direction , whenever you encounter a 0 you retain in same direction , if you encounter a 1’s you have to change direction to right of current direction and change that 1 value to 0, you have to find out from which index you will leave the matrix at the end.

2. given a coordinate (x,y) in a matrix of 0’s , make all the diagonals element through (x,y) as 1 in that matrix.

3. Basics about run time and compile time polymorphism. (with examples)

4. what is the need of run time polymorphism if we already have compile time polymorphism? (with example)

5. write a program to allocate a 3D-array dynamically.

HR:

1. Introduction.

2. Project and my role.

3. why do you want join samsung if you are already placed ?

I Hope my experience will help you, and Best of Luck from my side. 🙂

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Samsung Interview Experience | Set 4 (Samsung Research Institute, Delhi)

I am an **off-campus** hire. Please find my interview experience below:

1) Initially two written test were conducted:

a. First is **aptitude (25 Arithmetic Aptitude + 25 Data Interpretation).** The questions were easy but time consuming. In this round, both speed and accuracy matters. We have to **solve around 35 questions to clear this test**. So, we have to be accurate in solving 35 questions. In this test there is no negative marking, so after solving 35 questions, you can mark all the remaining questions and you are through this round.

b. Second is the **technical test (20 Questions).** They give an option to choose from **C or C++**. In this test, you have to **solve 14 – 15 questions** to get through. In this round, there are no options; we have to write the output on the space provided.

After this round, I get the phone call after **15 days** that I am selected for the interview.

2) Three round of Interviews were taken:

a. First round was based on **Operating System** concepts. They asked about **semaphores, inter-process communication, scheduling algorithms etc**.

b. Second round was based on **Algorithms** questions. The questions were:

c. Third is the **HR round** where negotiation about salary is there. It was just the formality.

After a **week**, I got the mail from HR of the final offer.

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

First Come First Serve (FCFS)

•Jobs are executed on first come, first serve basis.

•It is a non-preemptive, pre-emptive scheduling algorithm.

•Easy to understand and implement.

•Its implementation is based on FIFO queue.

•Poor in performance as average wait time is high.

Shortest Job Next (SJN)

•This is also known as **shortest job first**, or SJF

•This is a non-preemptive, pre-emptive scheduling algorithm.

•Best approach to minimize waiting time.

•Easy to implement in Batch systems where required CPU time is known in advance.

•Impossible to implement in interactive systems where required CPU time is not known.

•The processer should know in advance how much time process will take

Priority Based Scheduling

•Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.

•Each process is assigned a priority. Process with highest priority is to be executed first and so on.

•Processes with same priority are executed on first come first served basis.

•Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Round Robin Scheduling

•Round Robin is the preemptive process scheduling algorithm.

•Each process is provided a fix time to execute, it is called a **quantum**.

•Once a process is executed for a given time period, it is preempted and other process executes for a given time period.

- Context switching is used to save states of preempted processes.

Multiple-Level Queues Scheduling

Multiple-level queues are not an independent scheduling algorithm. They make use of other existing algorithms to group and schedule jobs with common characteristics.

- Multiple queues are maintained for processes with common characteristics.
- Each queue can have its own scheduling algorithms.
- Priorities are assigned to each queue.

For example, CPU-bound jobs can be scheduled in one queue and all I/O-bound jobs in another queue. The Process Scheduler then alternately selects jobs from each queue and assigns them to the CPU based on the algorithm assigned to the queue.

Why do we need scheduling?

A typical process involves both I/O time and CPU time. In a uniprogramming system like MS-DOS, time spent waiting for I/O is wasted and CPU is free during this time. In multiprogramming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.

Objectives of Process Scheduling Algorithm

Max CPU utilization [Keep CPU as busy as possible]

Fair allocation of CPU.

Max throughput [Number of processes that complete their execution per time unit]

Min turnaround time [Time taken by a process to finish execution]

Min waiting time [Time a process waits in ready queue]

Min response time [Time when a process produces first response]

SQL | Views

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition.

Syntax:

```
CREATE VIEW view_name AS

SELECT column1, column2.....

FROM table_name

WHERE condition;
```

view_name: Name for the View
table_name: Name of the table
condition: Condition to select rows

Read-Only VIEW

We can create a view with read-only option to restrict access to the view.

Syntax to create a view with Read-Only Access

```
CREATE or REPLACE FORCE VIEW view_name AS
        SELECT column_name(s)
        FROM table_name
        WHERE condition WITH read-only;
```

The above syntax will create view for **read-only** purpose, we cannot Update or Insert data into read-only view. It will throw an **error**.

DBMS - Indexing

Previous Page
Next Page

We know that data is stored in the form of records. Every record has a key field, which helps it to be recognized uniquely.

Indexing is a data structure technique to efficiently retrieve records from the database files based on some attributes on which the indexing has been done. Indexing in database systems is similar to what we see in books.

Indexing is defined based on its indexing attributes. Indexing can be of the following types −

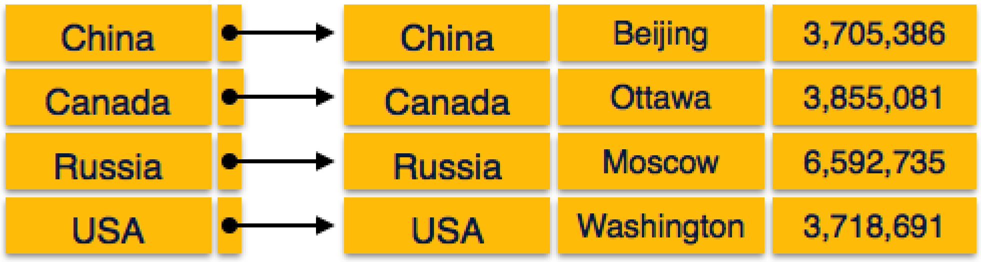
- **Primary Index** − Primary index is defined on an ordered data file. The data file is ordered on a **key field**. The key field is generally the primary key of the relation.
- **Secondary Index** − Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.
- **Clustering Index** − Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

Ordered Indexing is of two types −

- Dense Index
- Sparse Index

Dense Index

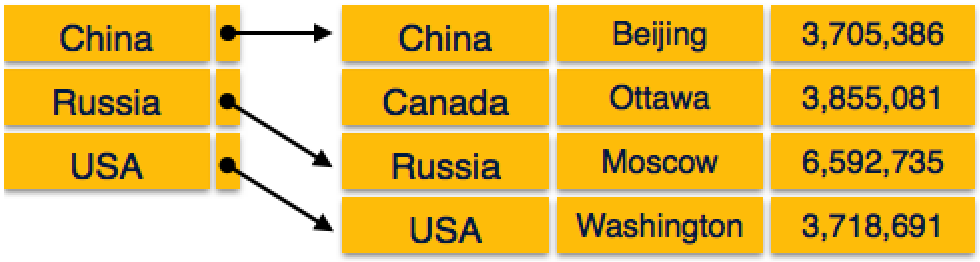
In dense index, there is an index record for every search key value in the database. This makes searching faster but requires more space to store index records itself. Index records contain search key value and a pointer to the actual record on the disk.



China	China	Beijing	3,705,386
Canada	Canada	Ottawa	3,855,081
Russia	Russia	Moscow	6,592,735
USA	USA	Washington	3,718,691

Sparse Index

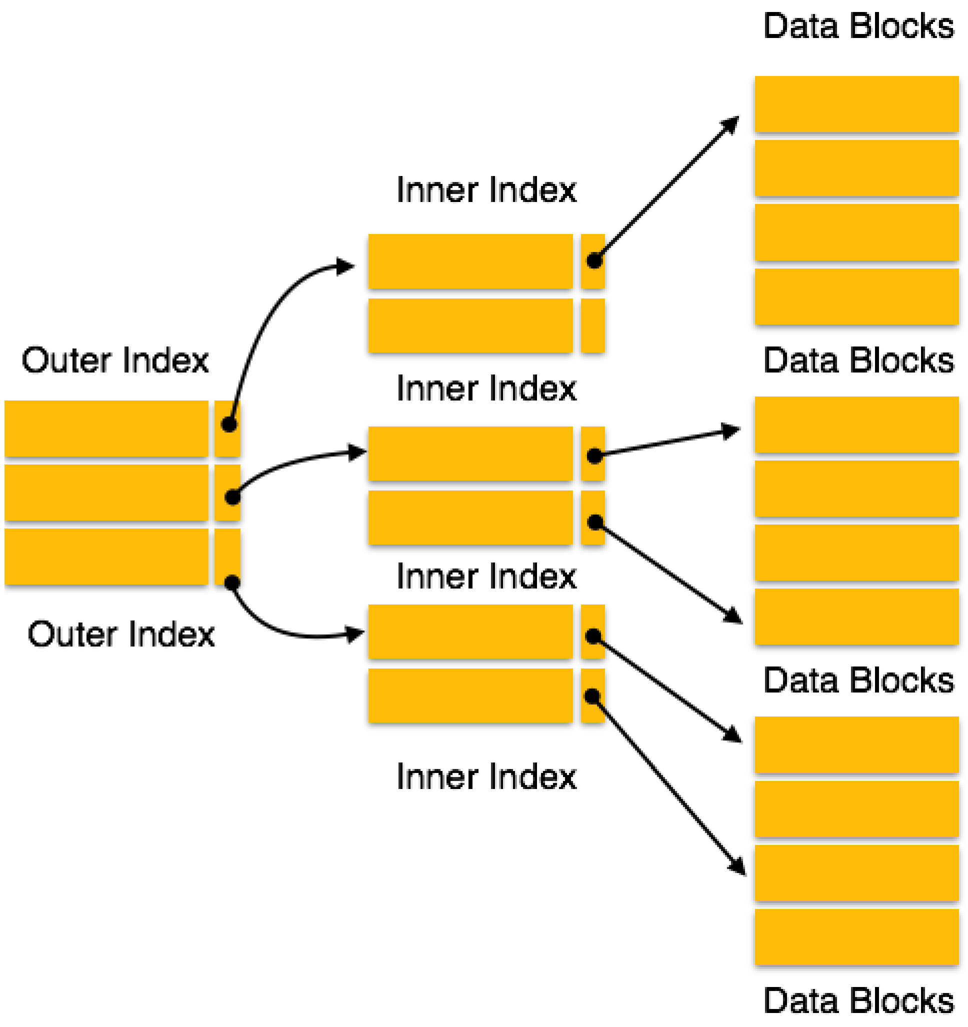
In sparse index, index records are not created for every search key. An index record here contains a search key and an actual pointer to the data on the disk. To search a record, we first proceed by index record and reach at the actual location of the data. If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.



China	China	Beijing	3,705,386
Russia	Canada	Ottawa	3,855,081
USA	Russia	Moscow	6,592,735
	USA	Washington	3,718,691

Multilevel Index

Index records comprise search-key values and data pointers. Multilevel index is stored on the disk along with the actual database files. As the size of the database grows, so does the size of the indices. There is an immense need to keep the index records in the main memory so as to speed up the search operations. If single-level index is used, then a large size index cannot be kept in memory which leads to multiple disk accesses.



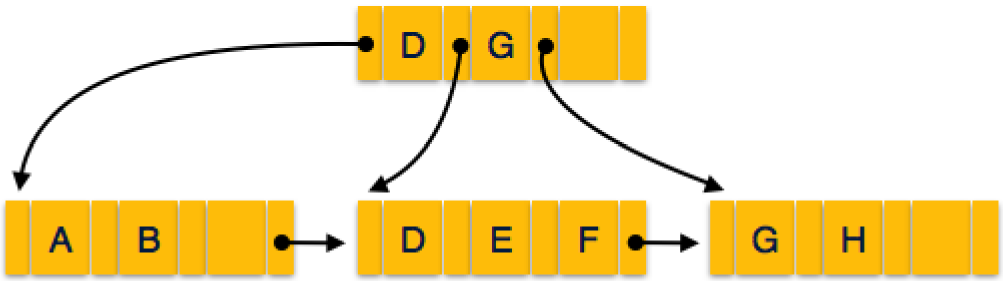
Multi-level Index helps in breaking down the index into several smaller indices in order to make the outermost level so small that it can be saved in a single disk block, which can easily be accommodated anywhere in the main memory.

B+ Tree

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the order **n** where **n** is fixed for every B+ tree.



Internal nodes −

- Internal (non-leaf) nodes contain at least ⌈n/2⌉ pointers, except the root node.
- At most, an internal node can contain **n** pointers.

Leaf nodes −

- Leaf nodes contain at least ⌈n/2⌉ record pointers and ⌈n/2⌉ key values.
- At most, a leaf node can contain **n** record pointers and **n** key values.
- Every leaf node contains one block pointer **P** to point to next leaf node and forms a linked list.

B+ Tree Insertion

- B+ trees are filled from bottom and each entry is done at the leaf node.
- If a leaf node overflows −
    - Split node into two parts.
    - Partition at **i = ⌊(m+1)/2⌋.**
    - First **i** entries are stored in one node.
    - Rest of the entries (i+1 onwards) are moved to a new node.
    - ***ith*** key is duplicated at the parent of the leaf.
- If a non-leaf node overflows −
    - Split node into two parts.
    - Partition the node at **i = ⌈(m+1)/2⌉**.
    - Entries up to **i** are kept in one node.
    - Rest of the entries are moved to a new node.

B+ Tree Deletion

- B+ tree entries are deleted at the leaf nodes.
- The target entry is searched and deleted.
    - If it is an internal node, delete and replace with the entry from the left position.
- After deletion, underflow is tested,

    - •If underflow occurs, distribute the entries from the nodes left to it.
- •If distribution is not possible from left, then
    - •Distribute from the nodes right to it.
- •If distribution is not possible from left or from right, then
    - •Merge the node with left and right to it.

DBMS - Indexing

Previous Page
Next Page

We know that data is stored in the form of records. Every record has a key field, which helps it to be recognized uniquely.

Indexing is a data structure technique to efficiently retrieve records from the database files based on some attributes on which the indexing has been done. Indexing in database systems is similar to what we see in books.

Indexing is defined based on its indexing attributes. Indexing can be of the following types −

- •**Primary Index** − Primary index is defined on an ordered data file. The data file is ordered on a **key field**. The key field is generally the primary key of the relation.
- •**Secondary Index** − Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.
- •**Clustering Index** − Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

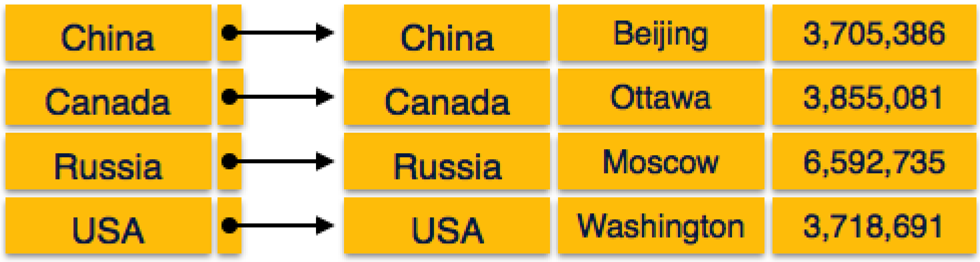
Ordered Indexing is of two types −

- •Dense Index
- •Sparse Index

Dense Index

In dense index, there is an index record for every search key value in the database. This makes searching faster but requires more space to store index records itself.

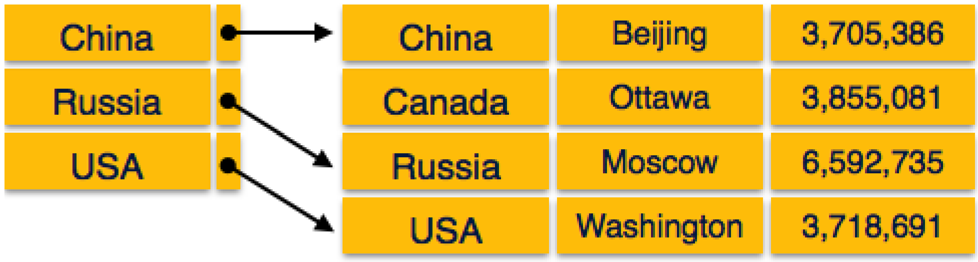
Index records contain search key value and a pointer to the actual record on the disk.



China	→	China	Beijing	3,705,386
Canada	→	Canada	Ottawa	3,855,081
Russia	→	Russia	Moscow	6,592,735
USA	→	USA	Washington	3,718,691

Sparse Index

In sparse index, index records are not created for every search key. An index record here contains a search key and an actual pointer to the data on the disk. To search a record, we first proceed by index record and reach at the actual location of the data. If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.



China	→
Russia	→
USA	→

China	Beijing	3,705,386
Canada	Ottawa	3,855,081
Russia	Moscow	6,592,735
USA	Washington	3,718,691

Multilevel Index

Index records comprise search-key values and data pointers. Multilevel index is stored on the disk along with the actual database files. As the size of the database grows, so does the size of the indices. There is an immense need to keep the index records in the main memory so as to speed up the search operations. If single-level index is used, then a large size index cannot be kept in memory which leads to multiple disk accesses.

Mutex:

In computer programming, a mutex (mutual exclusion object) is a program object that is created so that multiple program thread can take turns sharing the same resource, such as access to a file. Typically, when a program is started, it creates a mutex for a given resource at the beginning by requesting it from the system and the system returns a unique name or ID for it. After that, any thread needing the resource must use the mutex to lock the resource from other threads while it is using the resource. If the mutex is already locked, a thread needing the resource is typically queued by the system and then given control when the mutex becomes unlocked (when once more, the mutex is locked during the new thread's use of the resource).

General Questions:

1. Can a thread acquire more than one lock (Mutex)?

Yes, it is possible that a thread is in need of more than one resource, hence the locks. If any lock is not available the thread will wait (block) on the lock.

2. Can a mutex be locked more than once?

A mutex is a lock. Only one state (locked/unlocked) is associated with it. However, a recursive mutex can be locked more than once (POSIX complaint systems), in which a count is associated with it, yet retains only one state (locked/unlocked). The programmer must unlock the mutex as many number times as it was locked.

3. What happens if a non-recursive mutex is locked more than once.

Deadlock. If a thread which had already locked a mutex, tries to lock the mutex again, it will enter into the waiting list of that mutex, which results in deadlock. It is because no other thread can unlock the mutex. An operating system implementer can exercise care in identifying the owner of mutex and return if it is already locked by same thread to prevent deadlocks.

4. Are binary semaphore and mutex same?

No. We suggest to treat them separately, as it is explained signalling vs locking mechanisms. But a binary semaphore may experience the same critical issues (e.g. priority inversion) associated with mutex. We will cover these in later article.

A programmer can prefer mutex rather than creating a semaphore with count 1.

5. What is a mutex and critical section?

Some operating systems use the same word critical section in the API. Usually a mutex is costly operation due to protection protocols associated with it. At last, the objective of mutex is atomic access. There are other ways to achieve atomic access like disabling interrupts which can be much faster but ruins responsiveness. The alternate API makes use of disabling interrupts.

6. What are events?

The semantics of mutex, semaphore, event, critical section, etc… are same. All are synchronization primitives. Based on their cost in using them they are different. We should consult the OS documentation for exact details.

7. Can we acquire mutex/semaphore in an Interrupt Service Routine?

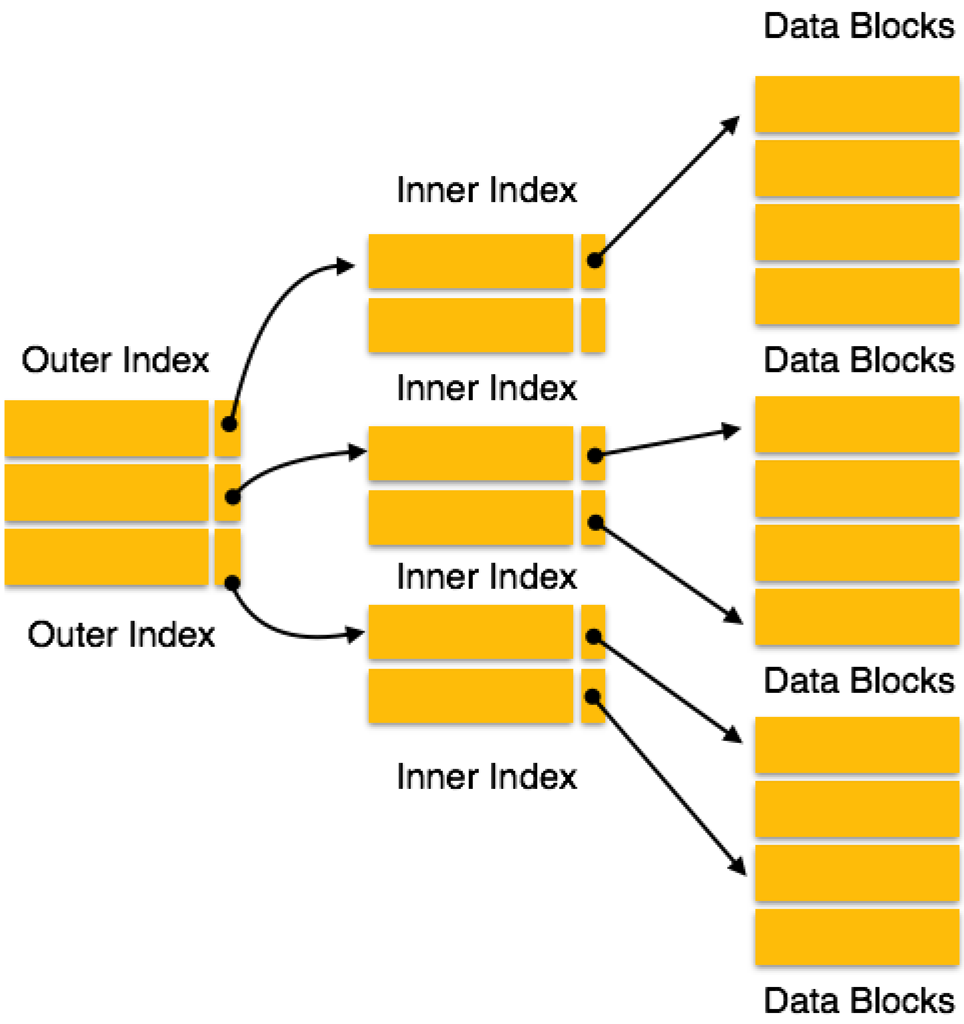
An ISR will run asynchronously in the context of current running thread. It is not recommended to query (blocking call) the availability of synchronization primitives in an ISR. The ISR are meant be short, the call to mutex/semaphore may block the current running thread. However, an ISR can signal a semaphore or unlock a mutex.

8. What we mean by “thread blocking on mutex/semaphore” when they are not available?

Every synchronization primitive has a waiting list associated with it. When the resource is not available, the requesting thread will be moved from the running list of processor to the waiting list of the synchronization primitive. When the resource is available, the higher priority thread on the waiting list gets the resource (more precisely, it depends on the scheduling policies).

9. Is it necessary that a thread must block always when resource is not available?

Not necessary. If the design is sure ‘what has to be done when resource is not available‘, the thread can take up that work (a different code branch). To support application requirements the OS provides non-blocking API.



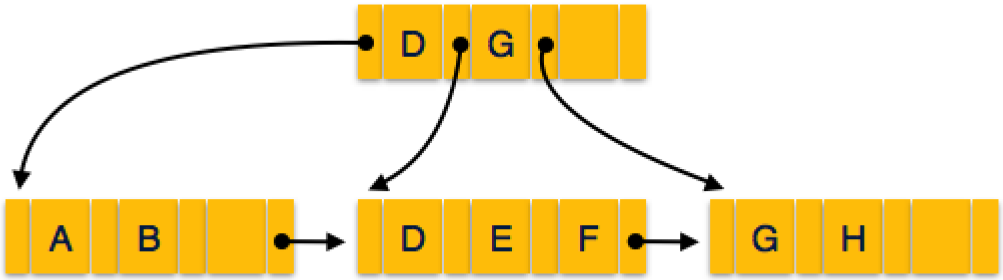
Multi-level Index helps in breaking down the index into several smaller indices in order to make the outermost level so small that it can be saved in a single disk block, which can easily be accommodated anywhere in the main memory.

B+ Tree

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the order **n** where **n** is fixed for every B+ tree.



Internal nodes −

- Internal (non-leaf) nodes contain at least ⌈n/2⌉ pointers, except the root node.
- At most, an internal node can contain **n** pointers.

Leaf nodes −

- Leaf nodes contain at least ⌈n/2⌉ record pointers and ⌈n/2⌉ key values.
- At most, a leaf node can contain **n** record pointers and **n** key values.
- Every leaf node contains one block pointer **P** to point to next leaf node and forms a linked list.

B+ Tree Insertion

- B+ trees are filled from bottom and each entry is done at the leaf node.
- If a leaf node overflows −
  - Split node into two parts.
  - Partition at **i = ⌊(m+1)/2⌋.**
  - First **i** entries are stored in one node.
  - Rest of the entries (i+1 onwards) are moved to a new node.
  - ***ith*** key is duplicated at the parent of the leaf.
- If a non-leaf node overflows −
  - Split node into two parts.
  - Partition the node at **i = ⌈(m+1)/2⌉**.
  - Entries up to **i** are kept in one node.
  - Rest of the entries are moved to a new node.

B+ Tree Deletion

- B+ tree entries are deleted at the leaf nodes.
- The target entry is searched and deleted.

    - If it is an internal node, delete and replace with the entry from the left position.
- After deletion, underflow is tested,
    - If underflow occurs, distribute the entries from the nodes left to it.
- If distribution is not possible from left, then
    - Distribute from the nodes right to it.
- If distribution is not possible from left or from right, then
    - Merge the node with left and right to it.

Critical Section: a critical section is group of instructions/statements or region of code that need to be executed atomically (read this post for atomicity), such as accessing a resource (file, input or output port, global data, etc.).

Remainder Section: The remaining portion of the program excluding the Critical Section.

Race around Condition: The final output of the code depends on the order in which the variables are accessed. This is termed as the race around condition.

In concurrent programming, if one thread tries to change the value of shared data at the same time as another thread tries to read the value (i.e. data race across threads), the result is unpredictable.The access to such shared variable (shared memory, shared files, shared port, etc…) to be synchronized.

A simple solution to critical section can be thought as shown below,

```
acquireLock();
Process Critical Section
releaseLock();
```

A thread must acquire a lock prior to executing critical section. The lock can be acquired by only one thread. There are various ways to implement locks in the above pseudo code.

A solution for the critical section problem must satisfy the following three conditions:

1. **Mutual Exclusion:** If a process Pi is executing in its critical section, then no other process is allowed to enter into the critical section.

2. **Progress:** If no process is executing in the critical section, then the decision of a process to enter a critical section cannot be made by any other process that is executing in its remainder section. The selection of the process cannot be postponed indefinitely.

3. **Bounded Waiting:** There exists a bound on the number of times other processes can enter into the critical section after a process has made request to access the critical section and before the requested is granted.