一共四轮，每轮都花二十五分钟问BQ。

1. OO design linux find command. 题目要求和讨论在https://leetcode.com/discuss/int ... -Linux-Find-Command。建议把所有的讨论都看一遍，里面有follow up questions。

2. 实现rate limiter.

3. system design. user can upload videos and watch videos. 需要考虑CDN, scalibilty, durability, load balance, cache.

4. leetcod 漆散尔。但是被包装成发电用电。

<https://leetcode.com/discuss/interview-question/369272/Amazon-or-Onsite-or-Linux-Find-Command>

I have my amazon interview coming up for SDE in seattle. One of my friends recently went on-site in Seattle and was asked this question in OOD. I have not been able to find a good approach to this question. Any suggestions would be helpful

implemnet linux find command as an api ,the api willl support finding files that has given size requirements and a file with a certain format like

1. find all file >5mb
2. find all xml  
   Assume file class  
   {  
   get name()  
   directorylistfile()  
   getFile()  
   create a library flexible that is flexible  
   Design clases,interfaces.
3. class File {
4. String name;
5. int size;
6. int type;
7. boolean isDirectory;
8. File[] children;
9. }
10. abstract class Filter {
11. abstract boolean apply(File file);
12. }
13. class MinSizeFilter extends Filter {
14. int minSize;
15. public MinSizeFilter(int minSize) {
16. this.minSize = minSize;
17. }
18. @Override
19. boolean apply(File file) {
20. return file.size > minSize;
21. }
22. }
23. class TypeFilter extends Filter {
24. int type;
25. public TypeFilter(int type) {
26. this.type = type;
27. }
28. @Override
29. boolean apply(File file) {
30. return file.type == type;
31. }
32. }
33. class FindCommand {
34. public List<File> findWithFilters(File directory, List<Filter> filters) {
35. if (!directory.isDirectory) {
36. return new NotADirectoryException();
37. }
38. List<File> output = new ArrayList<>();
39. findWithFilters(directory, filters, output);
40. return output;
41. }
42. private void findWithFilters(File directory, List<Filter> filters, List<File> output) {
43. if (directory.children == null) {
44. return;
45. }
46. for (File file : directory.children) {
47. if (file.isDirectory) {
48. findWithFilters(file, filters, output);
49. } else {
50. boolean selectFile = true;
51. for (Filter filter : filters) {
52. if (!filter.apply(file)) {
53. selectFile = false;
54. }
55. }
56. if (selectFile) {
57. output.add(file);
58. }
59. }
60. }
61. }
62. }