BQ(14点leadership principle 相关问题多过几遍): https://www.linkedin.com/pulse/how-interview-amazon-leadership-david-anderson/

1. 先解释项目, 项目怎么展开的, 项目中遇到了哪些困难, 项目中技术trade off, lead project

2. 以往有限时间内必须拿出结果的经历How to work with limited time or resource?

/赶deadline/有没有什么规划好却没有实现的功能，因为时间不够而放弃的项目里的本应实现的feature等等/要我說一個要compromise小細節來趕上的deadline的經驗/在之前的project中有没有遇到什么意外/ deadline 过后有没有做什么 improvement，怎么做/另一个是如果项目进度很慢怎么办tight deadline/讲一个被困难stuck住的经历

3. 为什么选了某个特定的技术/需要很快时间学习一个新知识来完成的项目/有没有为了project深入理解package运作的经验/dive in/遇到新问题且无法搜集相关资料怎么办/问project里为什么用某个技术，当时有没有调查别的可用的方法/你事先没有发现问题的例子

4.最自豪的经历/most interesting/most challenging / favorite project/ talk about project你想做而不是出于学业／工作压力做的？

5.现在的学习目标或者工作目标： 答： 多学点技术，准备工作 掌握基础知识， 之后打算自己想搞开发，研究新商业模式

6.Tell me about a time when you had difficulty working with someone on a side or class project? What did you do to overcome this?/ 受到强烈质疑该怎么办

7. why Amazon?

8. what’s the biggest mistake/failure you’ve made?/

9.举一个你没有遵守承诺的例子/ experience of not meeting the deadline./ 讲述一次放弃的经历

10. Describe a situation in which you took extra responsibility for a task that does not belong to you..

11. courses in this semester and plan courses next semester; 上过多少算法课

12.get feedback from others.然后问了我简历上的project，问我具体是怎么实现的，做出来的产品能干什么，作为用户该怎么用。然后问了一些BQ。有没有帮助别人的经历，有没有deadline来不及的经历（我说有为了更好地满足requirement，推迟过deadline。然后他紧接着问我deadline之后的requirement得到了多大的提升，我就从之前的事情里举了一个例子给他），有没有遇到过挫折以及是怎么克服挫折的。面试官一直在问project里的一个问题的安全性怎么解决，我说哇我当时没考虑到。是have you done object - oriented programming?

13. 然后问快速决定的经历，学到了哪些东西

14. 有没有承诺过做什么但没完成，有了新想法怎么开展，会不会跟老板讨论之类的。

15. what's the biggest mistake? how would you go about it if you don't have other resources and you have to get things down independently? what decision did you make without consulting others?

16. 有没有做过超出课业或项目要求的事情，在很短的时间内完成一件事情的经历

17. give me an example that you show ownership(后面接了一段ownership的定义），   
what do you do when the release date of your project is due and only 90% of the function is working（14 条里面的frugality, deliver results,  
这里面可能有同学会疑惑为什么是frugality， 我的理解是，与其浪费大量资源去寻求完美，不如先尽力做好90%，剩下的可以后面再完善。接受不完美非常重要)

18. 如何开始一个new project？遇到分歧如何处理？快到ddl了怎么办？距离ddl还有很长时间你会做什么

19. when you were assigned a project your proactive [att](http://https/redirect.viglink.com?key=a1aa544c3b328def412653f9fc432107&u=https%3A%2F%2Fwww.att.com%2Fshop%2Fwireless%2Fdevices%2Fcellphones.html)itude

20. 介绍一个在某项目中有一个坑比队友的经历，要求detail（项目、当时情况和你怎么做的）

20. 如何处理due很多的情况 - 时间不够你是否会放弃quality  
实习的project 你怎么做到写的保质保量，规划进度 - 后续组里面的人能继续重用你的代码吗

十四条圣经没有条条都准备case，因为经历实在太少削尖了脑袋也想不出了。。。但是感觉最重要的还是要把这些case连成一个你自己的故事，有连贯性，做到心中有数，这样在被问到没有准备的问题的时候，可以稍加润色就回答出来。

Some general questions:  
  
***Why***[***Amazon***](http://www.amazon.com/b?_encoding=UTF8&tag=1p3a-guanlian-20&linkCode=ur2&linkId=89c11e2c5b86155c5422f19cca1e9880&camp=1789&creative=9325&node=5)***?***  
Because I know Amazon is a top internet retailing company with a strong focus on customer experience. And there are a lot of talented people in amazon creating amazing products to make people’s life easier. They are always showing strong ownership to their projects and they always want to make things perfect. Those spirits do touch my heart. I remember when I just started to lead smart log analyzer project, everything is new to me. And I’m so passionate about creating great services for customers to save their precious time and money. They could have people hired to sit there watching the visuals we provide, and trying to do treat hunting, to see if there are malicious events among them. But we want it to be more convenient. We want our system to take care of all the stuff automatically. It will detect the malicious stuff, evaluate them, report them and provide suggestions and solve them automatically. The customer only needs to do clicks on the website to make simple decisions based on the suggestions system provide and they can focus more on their own things. That is really similar with Quilt. Quilt also does a great job to help customers focus on their codes. It will take care of patching automatically for hosts. That’s really amazing for customer. So, we have similar spirits. That’s one big reason why I want to join Amazon.   
Secondly, the growth of amazon is so impressive, it is not only an online retail shop, it also has a wide range of products like Alexa, [kindle](http://amzn.to/2PcJxHB), fire tablet and TV, Amazon web service, etc. There are so many fantastic technologies and products here. I will never feel boring if I am in amazon. And I’ll be excited about all the amazing things around me everyday. I can learn a lot of new technologies here. Beside that, I can also learn to work with solutions with full-stack point of view, by engaging requirements, front-end, back-end, storage and all the other stuff. I can understand the problems and solutions thoroughly.  
  
  
***Why you choose this position?***  
The job I applied is Software development engineer in Amazon web service quilt team. The reason why I choose this job is I have the similar experience. I remember when I just started to lead smart log analyzer project, everything is new to me. And I’m so passionate about creating great services for customers to save their precious time and money. They could have people hired to sit there watching the visuals we provide, and trying to do treat hunting, to see if there are malicious events among them. But we want it to be more convenient. We want our system to take care of all the stuff automatically. It will detect the malicious stuff, evaluate them, report them and provide suggestions and solve them automatically. The customer only needs to do clicks on the website to make simple decisions based on the suggestions system provide and they can focus more on their own things. That is really similar with Quilt. Quilt also does a great job to help customers focus on their codes. It will take care of patching automatically for hosts. That’s really amazing for customer. So, we have similar spirits. That’s one big reason why I want to join Amazon.   
Another big reason is amazon. the growth of amazon is so impressive, it is not only an online retail shop, it also has a wide range of products like Alexa, kindle, fire tablet and TV, Amazon web service, etc. There are so many fantastic technologies and products here. I will never feel boring if I am in amazon. And I’ll be excited about all the amazing things around me everyday. I can learn a lot of new technologies here. Beside that, I can also learn to work with solutions with full-stack point of view, by engaging requirements, front-end, back-end, storage and all the other stuff. I can understand the problems and solutions thoroughly.  
  
  
  
  
*The biggest mistake you made and what did you learn from it? (earn trust, customer obsession)*  
Case 1: The biggest mistake I’ve made happened when I led the smartloganalyzer project for the first time. The original schedule is week, but I suddenly received a message from my manager says that he will show a smartloganalyzer demo to one customer after two days. And at that time I was working on building the alerting system, that will be an important part of the demo. But I paid too much [att](http://https//redirect.viglink.com?key=a1aa544c3b328def412653f9fc432107&u=https%3A%2F%2Fwww.att.com%2Fshop%2Fwireless%2Fdevices%2Fcellphones.html)ention on details, I want to make everything perfect. I didn’t notice that I don’t have enough time to finish all the details. Last day before the demo day, my manager pointed that out. I apologized to him and then we started to figure out if there is a way to keep the demo the same as what we want it to be, but sacrificing some backend performance that customer won’t notice, cause it is just a demo. After the demo, I can rewrite the code to meet my requirements. Fortunately, the demonstration was not affected in the end. But from this mistake, I learnt that details are definitely important, but I also need to pay attention to the whole schedule, I need to always keep good communication with my teammates when I have my plan. I need to make sure that my schedule won’t affect other’s schedule.  
  
Case 2: I remember that after we first delivered the 5.1 version of SLA to customer, which is the version that first with machine learning feature and alerting feature. But the customer reported that they can not see the alerts. It is very weird cause after all the checks, all the features are running smoothly in the dev machine, but when it comes to the sensor, the alerting part will fail. After tracking the logs and error message, I found that the alerting system was not able to start at the very beginning. It seems that it doesn’t have the proper environment to run. Then I noticed that all the python related stuff in the system are based on python 3, except elastalert, which is based on python 2. So I double checked the document, I didn’t document the detail in the documentation. So the sensor doesn’t have python 2 environment. It is definitely a silly mistake. I apologized to my manager and the customer. And we added the python 2 envrionment to a new sensor as soon as possible and sent it to customer again. So, after that, I learnt that I should pay attention to each small detail during the development process, and document them in time so that we won’t forget. I need to make sure that all the things I delivered are fully qualified.  
  
  
***Most challenging project***  
The most challenging project is my current smartloganaylzer project. It is challenging because I need to implement all the new features on my own. It requires me to build machine learning tools for network threat hunting, I haven’t done anything related to threat hunting before. And It also requires me to build RESTful APIs for the tools. I haven’t developed any RESTful API before either. It also requires me to build alerting system and But the time is not unlimited, so I started to devote more time on the project, even my private time. I googled a lot of papers related to threat hunting, like malicious url, dnstuneling, network behavior. When I came up with some ideas, I would discuss with my manager to make sure the plan is feasible. And then I learnt Go as quick as possible so that I can design and implement all the RESTful APIs on time. Finally, all the new features have been implemented on time and we received positive feedbacks from the customer. So through this process, I learnt that it is so important to be curious and keep learning, the more you read and learn, the more problems you can solve. The feeling of ownership is really really important, the product is just like your own child. So “I don’t know how to do it” will never be the excuse.. 1point3acres  
  
  
  
  
  
***Couldn’t finish tasks before deadline***  
I remember last time I was building the alerting system, my original schedule is one week, but my manager suddenly sent me an email said that he needs to show a demo to a customer two days later. One core feature of the demo is the alerting system. So I was asked to realize the function before two days deadline. If I still follow the original schedule, I definitely can not finish that. My original schedule is to let the alerting system generate rules automatically based on dynamic data. But I can not finish that in such a short time. So I figured out another temporary solution with my manager, is to make it a fake automation temporarily, to preset all the parameters and processes. If you run it, it will behave like generating rules automatically, so for the demo, the customer will have exactly the same experience. And I can also finish that before deadline. After that, I can make it real automation.  
If I couldn’t finish tasks before deadline, I will discuss with my colleagues, trying to figure out a way that can improve the efficiency and If necessary, I will use my private time to keep working on the task. After all, finishing the task with high quality as soon as possible is what we want. I’ll never sacrifice the customer experience or the quality of the product because of that. Customer experience is always the most important. We must make sure that the product we are gonna deliver is qualified. We can sacrifice our own time to try to finish the tasks. If we still can not finish the tasks, we will communicate with customers and related people, to tell them why and earn their trust. At the same time, we will try our best to finish the tasks as soon as possible.  
  
**1. Customer Obsession**  
**Leaders start with the customer and work backward. They work vigorously to earn and keep customer trust. Although leaders pay attention to competitors, they obsess over customers.**  
. check 1point3acres for more.  
Case 1: When I led the smartloganalyzer project at the first time, it is very challenging to me. Cause I have no idea how to build the threat hunting structure. In order to understand it better, I googled a lot of papers and learnt a lot of basic concepts about network threat hunting. Based on that, I started to work with customers to figure out what kind of problems that customers concerned the most. Cause most customers who are using our products are not professional network developer. We need to use their language to tell them the story. They just need to know if they are under specific threats, they don’t care about how we find the threats. All they need to do is just make easy  decisions based on the suggestions we provide. The system will take care of all the other stuff. Like if there are brute force password attacking, the system will bubble up an alert and send an email to customer to let them know, and also providing them the links to the detailed dashboards. They can choose the suggestions provided by system, like putting the ip into the banlist.  
So, finally, after the delivery, we get positive feedbacks from customers. We are always trying our best to listen to our customers and provide the best experience for them.  
  
• Who was your most difficult customer?  
• Give me an example of a time when you did not meet a client’s expectation. What happened, and how did you attempt to rectify the situation?  
• When you’re working with a large number of customers, it’s tricky to deliver excellent service to them all. How do you go about prioritizing your customers’ needs?  
• Tell the story of the last time you had to apologize to someone.  
  
**2. Ownership**  
**Leaders are owners. They think long term and don’t sacrifice long-term value for short-term results. They act on behalf of the entire company, beyond just their own team. They never say “that’s not my job.**  
  
Case 1: When I worked on the SLA project, I was required to deploy elastalert and create few rules for the alerting system. Our alerting system is based on various elasticsearch indexes, and the situations and requirements for each index are definitely different. Since I was the person who was leading SLA project, I want my customers to always have the best experience. So I started to think about automate the generation. After the agreement from my manager, I categorized the current rules and the future rules based on different scenarios and requirements. Then I wrote a bunch of shell scripts to do the automation. I also created the APIs for those alerts, so if the customer doesn’t want the rule to be enabled, he can just click on website to turn it off. After delivery, we had strong positive feedbacks from customer.  
. From 1point 3acres bbs  
• Tell me about a time when you had to leave a task unfinished.  
• Tell me about a time when you had to work on a project with unclear responsibilities.  
  
  
**3. Invent and Simplify**  
**Leaders expect and require innovation and invention from their teams and always find ways to simplify. They are externally aware, look for new ideas from everywhere, and are not limited by “not invented here”. As we do new things, we accept that we may be misunderstood for long periods of time.**  
  
Case 1: When I worked at CCG, we have a ticketing system to record the development process, keep tracking all the tasks, bugs. But you know the ticketing system is a little bit crowded. It only shows a long list of tickets. With the first glance, you can hardly know current process of development. So to simplify it, I used a vis timeline to show all the tickets. Then we can see the whole process clearly through the timeline. With the timeline, we can schedule the tasks more easily. Everyone is happy with that.  
. check 1point3acres for more.  
• Tell me about a time when you gave a simple solution to a complex problem.  
• Tell me about a time when you invented something.  
  
-baidu 1point3acres  
**4. Are Right, A Lot**  
**Leaders are right a lot. They have strong judgment and good instincts. They seek diverse perspectives and work to disconfirm their beliefs.**  
  
Case 1: When I was building the alerting system through elastalert, I was asked to create rules for monitoring all the processes in the system. I have my initial plan to track the unique value of process names based on different sensors. But I don’t quite understand the behavior of system index, so I discussed with my manage, and he disagreed with plan. Actually we want to see the alert when the system is down, or nearly down. What we really care about is the count of processes. We want to alert when there is a significant drop in process numbers. So, after that I changed my design and everyone is happy.  
  
• Tell me about a time when you were wrong.  
• Tell me about a time when you had to work with incomplete data or information.  
  
  
**5. Learn and Be Curious**  
**Leaders are never done learning and always seek to improve themselves. They are curious about new possibilities and act to explore them.**  
  
Case 1:   
  
• Tell me about a time when you influenced a change by only asking questions.. check 1point3acres for more.  
• Tell me about a time when you solved a problem through just superior knowledge or observation.  
My manager asked me if there’s a way to export and import kibana dashboards, visuals more easily. Cause through the GUI side, it has the number limit and it is hard to manipulate from back end. So I came up with an idea we can utilize the REST apis kibana and elasticsearch provide. We can write scripts to automate the loading process according to our own needs. I know that’s feasible. So finally I wrote the scripts and all the stuff in kibana can be exported and imported automatically in back end.  
  
**6. Hire and Develop The Best**  
**Leaders raise the performance bar with every hire and promotion. They recognize exceptional talent and will move them throughout the organization. Leaders develop leaders and take seriously their role in coaching others.  We work on behalf of our people to invent mechanisms for development like Career Choice.**  
  
Case 1: Few months ago, there was an intern assigned to my project. He was helping me build the alerting system. But I found that he gets lost about what we are doing and what he is going to do. So, I told him that it is not as complex as what he is thinking about. I explained the alerting concepts to him one by one and categorized the things what we are doing and what we are going to do. After he is getting comfortable of the whole environment and what I need him to do, I started to assign him the specific task. And he is doing pretty well. So when someone comes to a new environment, getting lost is quite normal. We need to have patience to make him get comfortable of the while environment. Otherwise the efficiency will be very low. And that will also help him get used to the working environment quickly and increase his enthusiasm and devote more to the work.  
  
• Tell me about a time when you mentored someone.  
  
**7. Insist on the Highest Standards**  
**Leaders have relentlessly high standards – many people may think these standards are unreasonably high. Leaders are continually raising the bar and driving their teams to deliver high-quality products, services, and processes. Leaders ensure that defects do not get sent down the line and that problems are fixed so they stay fixed.**  
  
Case 1: When I was building the alerting system, one of the important features is to generate alerting rules automatically. But I found that if the indexes are in a large scale, or we meet some edge cases,  the efficiency of the process will be very low. Cause we are generating a lot of duplicate rules. So I built the cache to story the previous status, if there are new alerts happening or new indexes, then the system will create corresponding new rules for them. Otherwise it will only update them or do nothing. Also if there are old rules that we do not need anymore, we will remove them. And I also created links for those rules, so that we can save some disk space. It turned out that the efficiency was improved a lot.  
  
• Tell me about a time when you couldn’t meet your own expectations on a project.  
• Tell me about a time when a team member didn’t meet your expectations on a project.  
  
  
**8. Think Big**  
**Thinking small is a self-fulfilling prophecy. Leaders create and communicate a bold direction that inspires results. They think differently and look around corners for ways to serve customers.**  
  
  
• Tell me about your proudest professional achievement.  
My proudest professional achievement is the SLA project I owned. I tried my best to make it look perfect and make my customers have best experience. I build the machine learning tools for network threat hunting, each network anomaly and malicious event will be reported to the system. And I also created the alerting system, so that we don’t need people to sit there monitoring the boring data and create rules by hands. The alerting system will do all the stuff automatically. Customers love that, cause it really saves them a lot of time and money.  
• Tell me about a time when you went way beyond the scope of the project and delivered.-baidu 1point3acres  
When the first version of SLA that with machine learning feature was delivered, I was starting to think about moving the machine learning platform to tensorflow. Cause one of our core features is to analyze the behavior of network. But each network has its own environment and their behaviors are definitely different, so we need to build specific judging rules and awarding rules to let it train itself. So that it can always have good performance regarding to each environment. And I’m still working on that.   
  
  
**9. Bias for Action**  
**Speed matters in business. Many decisions and actions are reversible and do not need extensive study. We value calculated risk taking.**  
  
• Describe a time when you saw some problem and took the initiative to correct it rather than waiting for someone else to do it.  
  
• Tell me about a time when you took a calculated risk.  
  
• Tell me about a time you needed to get information from someone who wasn’t very responsive. What did you do?  
  
  
  
**10. Frugality**  
**Accomplish more with less. Constraints breed resourcefulness, self-sufficiency, and invention.  There are no extra points for growing headcount, budget size or fixed expense.**  
  
• Tell me about a time when you had to work with limited time or resources.  
  
. 1point3acres  
**11. Earn Trust**  
**Leaders listen attentively, speak candidly, and treat others respectfully. They are vocally self-critical, even when doing so is awkward or embarrassing.  Leaders do not believe their or their team’s body odor smells of perfume.  They benchmark themselves and their teams against the best.**  
  
• What would you do if you found out that your closest friend at work was stealing?  
  
• Tell me about a time when you had to tell someone a harsh truth.  
  
  
**12. Dive Deep**  
**Leaders operate at all levels, stay connected to the details, audit frequently, and are skeptical when metrics and anecdote differ. No task is beneath them.**  
  
• Give me two examples of when you did more than what was required in any job experience.  
Create elastalert rules, not only the rules, but also automate them.  
Create a vis timeline to visual the tickets.  
  
  
**13. Have Backbone; Disagree and Commit**  
**Leaders are obligated to respectfully challenge decisions when they disagree, even when doing so is uncomfortable or exhausting. Leaders have conviction and are tenacious. They do not compromise for the sake of social cohesion. Once a decision is determined, they commit wholly.**  
  
• Tell me about a time when you did not accept the status quo.  
• Tell me about an unpopular decision of yours.  
• Tell me about a time when you had to step up and disagree with a team members approach.  
• If your direct manager was instructing you to do something you disagreed with, how would you handle it?  
  
  
  
**14. Deliver Results**  
**Leaders focus on the key inputs for their business and deliver them with the right quality and in a timely fashion. Despite setbacks, they rise to the occasion and never settle.**  
  
• By providing an example, tell me when you have had to handle a variety of assignments. Describe the results.  
• What is the most difficult situation you have ever faced in your life? How did you handle it?  
• Give me an example of a time when you were 75% of the way through a project, and you had to pivot strategy–how were you able to make that into a success story?

Data Sturcture:

1. 什么是LinkedList, 什么是HashMap，hashset hashmap的区别 collision怎么处理等等如果HashTable有Collision怎么解决。
2. 问我算法课上学了什么，我说了几个比如贪心，dp啥的，然后问我什么情况下用贪心解决问题，我自我感觉这块回答得不好，我没总结出来，因为之前没有特意准备过，就举了一个具体实际的问题和他详细说了一下怎么用贪心解决，最后我说其实贪心不是很好，因为解决的方法不像dp有通用性（可能一道题一个解法），而且还必须得证明他的可行性和最优性，很繁琐，然后他就没继续问下去。
3. java和javascript有什么区别，我感觉好像是国内的面试一样，我就回答了java多线程， js单线程；java编译语言，js脚本语言等等诸如此类，然后面试官说ok了所以可能需要知道java是如何实现多继承的，treemap和hashmap和hashset的区别及实现之类的，总之能想到的都看看吧，面试官很喜欢在你解题的时候对于你正在用的数据结构提问
4. array, arraylist, linked list 分别是什么, random access 的 time complexity
5. 什么是 graph, tree, binary tree, binary search tree,用什么 data structure 存 graph
6. overriding + overloading
7. hashtable binary search tree
8. list, tree, hashmap有什么区别，什么情况下用哪一个。
9. 描述一下你最喜欢的sorting algorithm。
10. list 和map的区别
11. object oriented language的特性
12. Abstract Class是什么，和普通的class比有什么区别。之后问什么是Polymorphism
13. 问数组和链表有什么不同 我就扯了一堆什么插入时间复杂度 存储形式不同之类的 总之就是毫无逻辑的说了一大堆,然后问了一下数组和链表 查找 和添加元素所需要的时间复杂度 ,之后问了一个特别奇怪的问题 说是要在一台内存较小的移动设备上存储大量数据 究竟是应该用数组还是链表 我当时直接懵了 后来胡乱答了一通 说用链表 不然的话本来内存就很小 用数组得连续分配内存 容易external fragmentation 面试官说cool 但是给我的感觉是我没有答出他想要的答案但是讲的很有道理的样子（事后经过同学提醒应该是用数组 链表得多存一个指针）
14. binary tree 和 bst的区别，分别在其中search的时间复杂度。
15. BST和二叉树区别、hashtable和BST查找复杂度、什么时候用BST什么时候用hashtable；
16. 上来一个白人大哥，表示刚从vacation回来，查了一下他十几年经验，上来先让我自我介绍，之后开始bq：如何handle deadline，有没有什么很proud的project，队友是不是都contribute，其中有一个人没有contribute你是怎么办的，怎么沟通的，自己怎么想的。这些我都是基于之前做的一个搜索引擎回答的，紧贴leadership principle。问了15分钟之后开始coding，结果，大哥说他打不开coding pad，他打不开我就打不开，结果……大哥说那你share一下你的screen给我看一下你之前做过的project吧。我？？？还可以这样？好在存了很多project在电脑，首先展示了machine learning一个decision tree和random forest (Java), 之后解释了怎么建立的类，怎么优化，怎么train和test model，什么是precision recall F-score；之后又展示了computer vision的shape from shade 3D reconstruction(Python)，大哥觉得that's cool，其中详细讲了一下如何利用matrix operation取避免loop，从而完成时间的优化。最后开始讨论distributed system。大哥一直念念不忘之前讲的搜索引擎的project，于是问，CAP theorem在这个project里面要怎么处理C和A的优先级，如果先处理A那么用户输入的query用于feedback的信息如何backup，leadership如何搞，如果先选C，如何优化用户体验；如果CAP放到bank atm design呢要怎么选？为什么？有哪些critical需要注意的点？我就这么和大哥侃了半个小时，到最后还扯到了NLP的一些点。大哥说这些问题都是比较难的，glad you were thinking of them while doing your project。
17. 说说对于queue和stack的区别；如何用queue实现stack
18. 就用了我BFS遍历图的时间复杂度问题
19. 问到了priority queue是怎么构建的，pop完这个pq怎么变化，还问了map的key-value是怎么对应的，我写的code里有用到排序还问了几种排序算法，基本就是我说到可以用什么什么他就会问是怎么实现的什么结构。
20. 第一题是个数据结构题，叫我如何实现一个BST并且如何实现曾删改查，还问了各种操作的时间复杂度。（在增加和删除后要维护BST的结构，不要忘了）之后由树转化到HashTable，问HashTable的原理，用冲突的解决方法，各自的优缺点。最后还问了hash函数怎么设计才是好的，这个楼主也不太会。。。就随便说了一下。
21. hashtable 作死说open addressing and chaining 被问到了底  
                    stack queue  
                    sorting merge sort的worst case 说错了（人傻了  
                    process and threading的大量问题  
                    linedlist和hashmap的区别
22. 1.heap是什么  
    2.heap存什么的（动态变量）  
    引出stack是静态变量  
    问了个stack能做什么- lz想起ds的时候写的是计算器 用的stack  
    然后问了一些 heap、stack基础操作的时间复杂度 比如删除啥的
23. 深入问hashmap的实现，如何处理collision，用Linkedlist处理冲突的缺点是什么，能不能换个数据结构等。总之对hash table要掌握，见到过好多个问hash table的了。