Soft Skill Interview

4 인적성 면접

Soft Skill Interview

- 커뮤니케이션 - 전달하는 방법
- FAQ - 자주 묻는 질문
- STAR approach - 정리법

Communication Tips

- 숫자
- 메타인지
- 말투와 온도

숫자를 활용하세요

- [심각성, 지속성, 긴급성]

숫자를 활용하세요

- 인턴 테스터로 유닛 테스트를 썼습니다
  - 심각성: 테스트 커버리지 15% -> 82%
  - 지속성: 메일 1시간의 반복업무를 자동화
  - 긴급성: 2주만에 달성

메타인지

- 아는 것과 모르는 것의 구분

메타인지

- 아는 것과 모르는 것의 구분
  - Did that make sense?
  - DId I lose you?
  - Do you need more details?
  - Let me recap
  - Did i answer your question?
  - Could you elaborate more?

자기소개

- Please introduce yourself
- 상대방의 자기소개
- 년차와 기대치는 비례

주니어에게 요구되는 스킬

- Communication / Teamwork
- Overcome hardships
- Ownerships
- Meeting the deadline

주니어에게 요구되는 스킬

Communication / Teamwork	Tell me about a conflict faced at work and how you dealt with it
Overcome hardships	Tell me about a challenge faced at work and how you dealt with it
Ownerships	What is the most significant problem you solved in the workplace
Meeting the deadline	How do you prioritize your work? Have you ever missed a deadline

Amazon Leadership Principles

- Customer Obsession
- Ownership
- Invent and Simplify
- Are Right A Lot
- Learn and Be Curious
- Hire and Develop the Best
- Insist on the Highest Standards
- Think Big
- Bias For Action
- Frugality
- Earn Trust
- Dive Deep
- Have Backbone; Disagree and Commit
- Deliver Results
- Strive to be Earth’s Best Employer
- Success and Scale Bring Broad Responsibility

예시

- Tell me about a challenge or conflict faced at work and how you dealt with it

예시

- Tell me about a challenge or conflict faced at work and how you dealt with it

나는 갑자기 경험하지 못한 솔루션을 찾아야 했던 적이 있다. 시니어가 어떤 사유로 공석일때 혼자 문제를 주도적으로 해결한 경험을 공유해보겠다. 서버 수량이 모잘라서 더이상 클러스터를 띄울 수 없는 상황에 맞닥 뜨렸다. 처음엔 막막했지만, 삼일만에 해결했다. 사용량이 적은 클러스터 리스트를 뽑아 전달했고, 이 클러스터들을 통합하는 방식으로, 자원의 효율화를 통해 문제를 해결했다.

예시

- Tell me about a challenge or conflict faced at work and how you dealt with it

나는 갑자기 경험하지 못한 솔루션을 찾아야 했던 적이 있다. 시니어가 어떤 사유로 공석일때 혼자 문제를 주도적으로 해결한 경험을 공유해보겠다. 서버 수량이 모잘라서 더이상 클러스터를 띄울 수 없는 상황에 맞닥 뜨렸다. 처음엔 막막했지만, 삼일만에 해결했다. 사용량이 적은 클러스터 리스트를 뽑아 전달했고, 이 클러스터들을 통합하는 방식으로, 자원의 효율화를 통해 문제를 해결했다.

얼마나 효율화 했나? 안쓰는 클러스터는 없던가?

예시

- Tell me about a challenge or conflict faced at work and how you dealt with it

나는 갑자기 경험하지 못한 솔루션을 찾아야 했던 적이 있다. 시니어가 어떤 사유로 공석일때 혼자 문제를 주도적으로 해결한 경험을 공유해보겠다. 서버 수량이 모잘라서 더이상 클러스터를 띄울 수 없는 상황에 맞닥 뜨렸다. 처음엔 막막했지만, 삼일만에 해결했다. 사용량이 적은 클러스터 리스트를 뽑아 전달했고, 이 클러스터들을 통합하는 방식으로, 자원의 효율화를 통해 문제를 해결했다.

얼마나 효율화 했나? 안쓰는 클러스터는 없던가?

그 후 주기적으로 리스트를 뽑아 담당자에게 알림을 보내고, 통합 작업을 해서 서버 효율화를 정기적으로 진행했고, 사용량이 아주 적은 경우 반납을 받아 새로 꼭 필요한 부서에 리소스를 할당했다.

STAR Approach

- Situation
- Task
- Action
- Result

STAR Approach

Points	Situation	Task -> Problem	Action	Result
#Customer Obsession #Dive deep #Bias for Action	my most challenging customer issue. Customer is locked out of their system. No user can log in.	EUDP Limited Data	Instead of perfect solution. take action.	I could provide solutions for customers even when I could not have any access to their system and logs. Even when I do not have enough data to make decisions, I would look for alternative ways to figure out root causes, and resolve customers' issues in timely manner.

STAR Approach

- 스토리 갯수는 여유있게 준비
- 인재상이 있다면 종류별로 여러개 짚어주기
- 응급 상황에 대한 대본

인적성 면접 예시 1

I was in charge of a service that can run scripts against systems on cloud environment.
Scripts were written to check for status and property values for each system.
This service could collect the query results and record them in designated excel file.

인적성 면접 예시 1

Points	Situation	Task -> Problem	Action	Result
#Ownership #Earn Trust #Communication	I was in charge of a service that can run scripts against systems on cloud environment. Scripts were written to check for status and property values for each system. This service could collect the query results and record them in designated excel file.	Requests would fly in, and it was very tough that always the most important and critical queries come in the last minute.	Reduce redundant manual work by using schedulers. Implement UI to display query processes and results. Simplify manual input to minimize time Prepare documentations so that other colleagues can also process the requests	What is the best way to get help when you needed? You first help others when they need you.

인적성 면접 예시 1

FAQ

- Can you discuss a time where your team had to work through a difficult situation
- How do you explain new topics to coworkers unfamiliar with them?
- Describe a time when you faced conflict and how you addressed it
- Talk about a time when you made a point that you knew your colleagues would be resistant to

인적성 면접 예시 2

I suggested my manager if I should take over a service and become a service owner. Since this was the first time I was owning a service, he gave me one that is in maintenance mode. Even though it was running fine, I took another look to learn the service. I saw if there is any improvement I can make. I found out this service was using more memory than needed. I tried to save money by only using what services need and decreased memory for each instance of this service. Few weeks later, I got a call saying only half of the instances are successfully restarted. I could not find out why until it recovered itself. I continued investigating through that morning. Figured out it was due to platform update.

인적성 면접 예시 2

Points	Situation	Task -> Problem	Action	Result
##Ownership #Frugality #Dive Deep #Overcome Hardship	I suggested my manager if I should take over a service and become a service owner. Since this was the first time I was owning a service, he gave me one that is in maintenance mode.	I found out this service was using more memory than needed.	I tried to save money by only using what services need and decreased memory for each instance of this service I wanted to make sure that my service is running fine with the change. I kept an eye on next deployment cycle, and I confirmed that my changes are not causing any errors.	Even though its only mb scale difference: 1gb -> 64mb, 16 times smaller, it would make quite a bit of impact since its in cloud environment. 4 instances * n data centers.

인적성 면접 예시 2

Points	Situation	Task -> Problem	Action	Result
##Ownership #Frugality #Dive Deep #Overcome Hardship	Few weeks later, I got a call saying only half of the instances are successfully restarted. I got called in the early morning, and while I was investigating, it got deployed after an hour.	I could not find out why until it recovered itself.	I continued investigating through that morning.	Found out we had platform update. I wrote documentations about platform update schedules so that other service owners wont face the same issue.

인적성 면접 예시 2

FAQ

- Tell me about a situation where you had to make a decision without managerial supervision
- Tell me about a time when you took on something significant outside your area of responsibility. Why was it important? What was the outcome?
- Tell me about a time when you above and beyond
- What is the most significant problem you solved in workplace?

인적성 면접 예시 2

- Tell me about a situation where you had to make a decision without managerial supervision

I had a chance to make decisions all by myself as a service owner. I decided to reduce memories of service deployment properties because I could see the usage was way below 10%, nearly 5 % on average. When this decision led to a minor issue, even though it did not cause any significant problem in production, I went through thorough troubleshooting. As a result, I could write documentation to share with my team, about minimum recommended memory usage for service deployments, and warn the team about the importance of sharing platform update schedules.

인적성 면접 예시 2

- Tell me about a time when you took on something significant outside your area of responsibility. Why was it important? What was the outcome?

I once asked for extra work. I wanted to own a service, so I asked my manager for a service to own. Then I took initiatives to take a close look and see if there is any improvements I can make. I realized my service was wasting extra memories, so I decided to reduce the memory amount in the service deployment file to save resources. This decision had multiplying effects as in cloud platforms, services are deployed in multiple instances in multiple data centers.

인적성 면접 예시 2

- Tell me about a time when you above and beyond

I would like to share my story about how I dealt with my very first issue as a service owner. Even though the service recovered in an hour, I wanted to dive deep and figure out why half of the deployment instances took longer to be up and running. Even though it did not cause any significant problem in production, I went through thorough troubleshooting. As a result, I could write documentation to share with my team, about minimum recommended memory usage for service deployments, and warn the team about the importance of sharing platform update schedules.

인적성 면접 예시 2

- What is the most significant problem you solved in workplace?

I once had an issue where my service took an hour to have all instances up and running. Even though it did not cause any significant problem in production, I went through thorough troubleshooting. It was very hard to find because the issue was not caused by our side. I had to look through other services log and figured out that all of the services were triggered to restart at a certain time. I thought there must be a general issue with the platform for all services to be triggered restart at the same time. My guess was right, and I could prove it by asking about platform update schedules to the platform team. As a result, I could write documentation to share with my team, about minimum recommended memory usage for service deployments, and warn the team about the importance of sharing platform update schedules.

Coding Interview

5 코딩 면접

기술 인터뷰의 범위

- Object Oriented Programming
- System Design
- 기술 스택

코딩 인터뷰

- Coding Style
- Data Structure
  stack, list, queue, hashmap, set
- Complexity
  time / space
- Algorithm
  Graph theory, dynamic programming
- Communication Skill

답변 구성

1. Paraphrase
2. Input. Output. Scope. Range definition
3. Run examples (3 sets) and expected outputs
4. Edge cases
5. Work through the problem manually + discuss approach
6. Implement
7. Walk through your code
8. Fix bugs
9. Complexity
10. Potential improvements

예제 1

Given an array of `nums` and an `target`, return indices of the two numbers such that they add up to `target`.

예제 1

Given an array of nums and an target, return indices of the two numbers such that they add up to target.

Nums = 1,2,3,4 target = 3 -> ret = 0,1
Nums = 4,5,6,7 target = 10 -> ret = 0,2

예제 1

Given an array of nums and an target, return indices of the two numbers such that they add up to target.

Nums = 1,2,3,4 target = 3 -> ret =[0,1]
Nums = 4,5,6,7 target = 10 -> ret = [0,2]

Nums = [] target = 0 -> ret = [-1]
Nums = [1,2] target = 1 -> ret = [-1]

예제 1

Given an array of nums and an target, return indices of the two numbers such that they add up to target.

Nums = 1,2,3,4 target = 3 -> ret =[0,1]
Nums = 4,5,6,7 target = 10 -> ret = [0,2]

Nums = [] target = 0 -> ret = [-1]
Nums = [1,2] target = 1 -> ret = [-1]

```
// brute force

// for each index
        // loop through last items
                //sum and check if sum equals target
                // if so, return
//return [-1]
```

예제 1

```
Nums = 1,2,3,4 target = 3 -> ret =[ 0,1]
Nums = 4,5,6,7 target = 10 -> ret =  [0,2]

Nums = []  target = 0 -> ret = [-1]
Nums = [1,2] target = 1 -> ret = [-1]
```

```
// brute force

// for each index
        // loop through last items
                //sum and check if sum equals target
                // if so, return
//return [-1]
```

```
public int[] findSum(int[] nums, int target) {
    for(int i = 0; i < nums.length; i++){
        for(int j = i+1; j < nums.length; j++){
            if(nums[i]+nums[j] == target){
                return new int[] {i,j};
            }
        }
    }
    return new int[] {};
}
```

예제 1

```
Nums = 1,2,3,4 target = 3 -> ret =[ 0,1]
Nums = 4,5,6,7 target = 10 -> ret =  [0,2]

Nums = []  target = 0 -> ret = [-1]
Nums = [1,2] target = 1 -> ret = [-1]
```

```
// brute force

// for each index
        // loop through last items
                //sum and check if sum equals target
                // if so, return
//return [-1]
```

```
public int[] findSum(int[] nums, int target) {
    for(int i = 0; i < nums.length; i++){
        for(int j = i+1; j < nums.length; j++){
            if(nums[i]+nums[j] == target){
                return new int[] {i,j};
            }
        }
    }
    return new int[] {-1};
}
```

예제 1

```
public int[] findSum(int[] nums, int target) {
    for(int i = 0; i < nums.length; i++){
        for(int j = i+1; j < nums.length; j++){
            if(nums[i]+nums[j] == target){
                return new int[] {i,j};
            }
        }
    }
    return new int[] {-1};
}
```

예제 1

```
public int[] findSum(int[] nums, int target) {
   for(int i = 0; i < nums.length; i++){
      for(int j = i+1; j < nums.length; j++){
         if(nums[i]+nums[j] == target){
            return new int[] {i,j};
         }
      }
   }
   return new int[] {-1};
}
```

```
public int[] findSum(int[] nums, int target) {
   HashMap<Integer, Integer> map = new HashMap();
   for(int i = 0; i < nums.length; i++){
      int diff = target - nums[i];
      if(map.containsKey(diff)){
         int j = map.get(diff);
         if(j != i)
            return new int[] {j, i};
      }
      map.put(nums[i], i);
   }
   return new int[] {-1};
}
```

예제 2

Given the head of a singly linked list, return the middle node of the linked list.

예제 2

Given the head of a singly linked list, return the middle node of the linked list.

예제 2

Given the head of a singly linked list, return the middle node of the linked list.

```
public Node getMid(Node head) {

}
```

예제 2

Given the head of a singly linked list, return the middle node of the linked list.

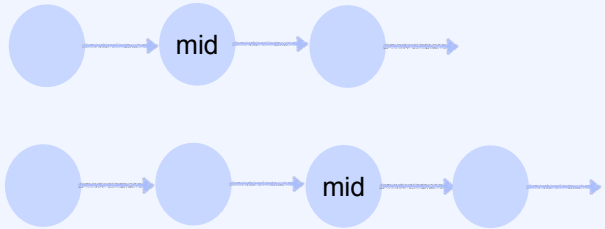
```
public Node getMid(Node head) {

}
```

```
class Node {
        Int val;
        Node next;
}
```

예제 2

Given the head of a singly linked list, return the middle node of the linked list.



예제 2

Given the head of a singly linked list, return the middle node of the linked list.

```
//cnt = 0;
//curr = head;
// until next != null
        // curr = curr.next
        // cnt++;
//for (int i = 0 ; i <= cnt /2; i++)
        //curr = curr.next
//return curr
```

```
// O(n)


//O(n/2)
```

예제 2

Given the head of a singly linked list, return the middle node of the linked list.

```
//cnt = 0;
//curr = head;
// until next != null
        // curr = curr.next
        // cnt++;
//for (int i = 0 ; i <= cnt /2; i++)
        //curr = curr.next
//return curr
```

```
//cnt = 0;
//fast, slow = head;
// until fast != null
        // fast = fast.next.next
        // slow = slow.next
//return slow
```

예제 2

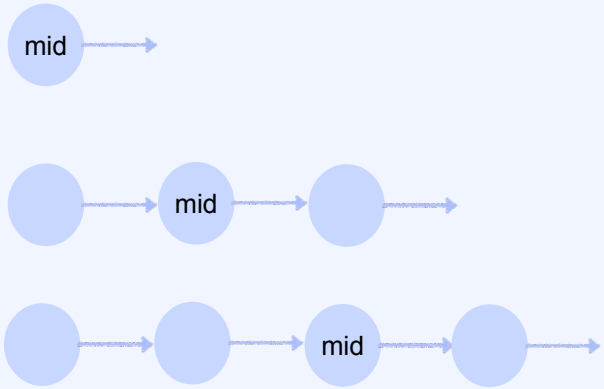
Given the head of a singly linked list, return the middle node of the linked list.

```
//cnt = 0;
//fast, slow = head;
// until fast != null
        // fast = fast.next.next
        // slow = slow.next
//return slow
```

```
public Node getMid(Node head) {
    Node slow = head;
    Node fast = head;
    while(fast != null){
        if(fast.next!= null){
            fast = fast.next.next;
            slow = slow.next;
        }else{
            return slow;
        }
    }
    return slow;
}
```

예제 2

Given the head of a singly linked list, return the middle node of the linked list.




```
public Node getMid(Node head) {
    Node slow = head;
    Node fast = head;
    while(fast != null){
        if(fast.next!= null){
            fast = fast.next.next;
            slow = slow.next;
        }else{
            return slow;
        }
    }
    return slow;
}
```

마무리

수고하셨습니다