

コーディングテスト体験記

横山研M2 北村

自己紹介

- AtCoderギリギリ水
- 就活終了

目次

- 概要
- 評価ポイント
- 問題傾向
- 出題内容各種
 - 初級アルゴリズム
 - 上級アルゴリズム
 - 知識問題
 - Web系実装
 - ゲーム系実装
 - その他

概要

コーディングテストの使われ方は以下

- おおよそ書類選考の後
- 適性検査(SPI)と同タイミング
 - つまり足切り目的
- 筆者が就活で受けた企業ではすべてで実施

→加点要素ではなく減点要素

評価ポイント

会社によるが見ているポイントは以下

- 解答の方針=思考力(正誤は問わない場合も)
- クラスや関数の切り方
- コーディング規約/命名規則等のお約束
- (点数しか見ていない場合)

面接担当のエンジニア等に細部までコードを読まれていることを念頭に！

人に見せるための一番キレイなコードを書く

問題傾向

問題構成

- 大抵は競プロ問題が3~5つ程度
- 各30分~2時間程度と時間は余裕
- 恐らく2~3問目(約300点相当)までは足切り、それ以降(約500点相当)は解けなくても通る

AtCoderと比べると実装が重め

- 複雑な条件分岐
- 露骨なコーナーケース
- (AC Libraryが使えないことに注意)

初級アルゴリズム

- 累積和
- 2分探索
- 素数判定
- 文字列操作
- GCD/LCM

足切りレベル・AtCoder300点相当の問題例

このレベルは解けないと厳しいので、上記のレベルで自信がない人は危険

上級アルゴリズム

- 優先度付きキュー
- DP
- bit全探索
- (☆重み付き)Union-Find
- グラフ各種

解けてなくても通ることがある他、貪欲な手法等でもらえる小さい制約の部分点があったりする例

このへんまでキレイに解けると満点取れてハッピー

知識問題

コードではなく選択問題が出る場合の例

- C++/Java/Python数行の簡単な読解
- SQLの読解
- 情報数学(bitの演算など)
- IT用語(“SaaS”, “AWS”など)
- 算数/数学の計算問題(中学・暗算レベル)

基本情報とかより1段階簡単なレベル。約20問で制限時間が10~20分くらいだったと思うので、調べる時間はほとんど無いことに注意

Web系実装

Web系のポジションの場合に出る問題

- 簡単なAPIの作成
- インタラクティブ問題でAPIを叩く
- SQL
- HTML/CSS

企業やポジションによってアルゴリズム知識の代わりにWebの知識が必要な問題が出ることも

アルゴリズム系と選択問題になっている場合もある

ゲーム系実装

ソフトウェアメーカーで見た問題

- ベクトル演算
- 3次元の衝突判定
- ☆空間分割

とにかく衝突判定が頻出なので3次元のベクトルクラスを用意する

テストは関係ないけどゲーム作品の提出が重視されてたのできちんと完成させた作品を持っていくとハッピー

その他

オンラインテストではなく長期間の課題

- RailsでWebアプリケーション全体を作成
- 1週間でゲーム作品を作成

データサイエンスのポジション

- pandas, numpyの取り扱い
- データの前処理(欠損値補完・one-hot encodingとか)

まとめ

- 人に読ませるためのキレイなコードを書く
- 実装重視なので早解き勢は注意
- AtCoder300点は解けないと危険
- 500点が解けるなら安心
- その他目指すポジションに応じたスキル