



Multilabel Questions Classification

Submitted By :

Sarvat Ali (2018201009)

Prabha Pandey (2018201053)

C. Sai Sukrutha (2018201054)

Shafiya Naaz (2018201062)

Submitted To :

Dr. Radhika Mamidi



Problem Statement

- ❑ Multilabel classification of questions from coding platforms like codechef, codeforces etc and labelling them based on their respective paradigms.



Motivation

- ❑ To facilitate the user to be able to access multiple websites at once and get all the questions related to the programming paradigm he is interested in.
- ❑ For helping the user to identify the category of question he/she is trying to solve.



Data Scraping

- ❑ Previously, we scraped out data from leetcode, but the data extracted did not have labels.
- ❑ Therefore we did data scraping from codechef and codeforces where the labels are provided.
- ❑ We have used the following libraries for scraping data :
 - ❑ requests==2.22.0
 - ❑ beautifulsoup4==4.8.0

Sample Dataset Structure

Codechef:

Sample From Data Set

QuestionCode	Title	QuestionLink	DifficultyLevel	Problem Statement	Editorial	Tags	Time Limit	Languages	Solution	SubmissionID
CONFLIP	Coin Flip	/problems/CONFLIP	easy	All submissions for this	http://disc	['khabarbasha', 'simple-math', 'ad-hoc', 'nov12', 'cakewalk']	0.09	C	#include<stdio.h>	S9890254
CONFLIP	Coin Flip	/problems/CONFLIP	easy	All submissions for this	http://disc	['khabarbasha', 'simple-math', 'ad-hoc', 'nov12', 'cakewalk']	0.54	C++ 4.3.2		S9890620

Sample Dataset Structure

Codeforces:

Sample From Data Set

id	name	problem statement	tags	difficulty	solution	time_taken	author
1143C	Queen	You are given a rooted tree	dfs and similar		using namespace std; int main(){	46 ms	munjalvaibh
1143B	Nirvana	Kurt reaches nirvana when	brute force,math,number theory		#include <bits/stdc++.h> using namespace std; int f(int n){ if(n<10)return max(1,n);	31 ms	Andrija
1143A	The Doors	Three years have passed	implementation		#include <bits/stdc++.h>	155 ms	179000
1142D	Foreigner	Traveling around the world	dp		no code found	NA	NA



Data Preprocessing

- ❑ The scraped data is stored in a csv file, which is further used in data preprocessing to make it usable.
- ❑ Following are the steps performed for cleaning up of data set :
 - ❑ Removing Stop Words
 - ❑ Removing Special Characters
 - ❑ Applying Stemming (as per English Dictionary)
 - ❑ The above approaches are not applied on solution.

Data Preprocessing continued..

- ❑ The data scraped had some unused tags, so we removed the data items labeled by any of those tags.



Un-Used Tags

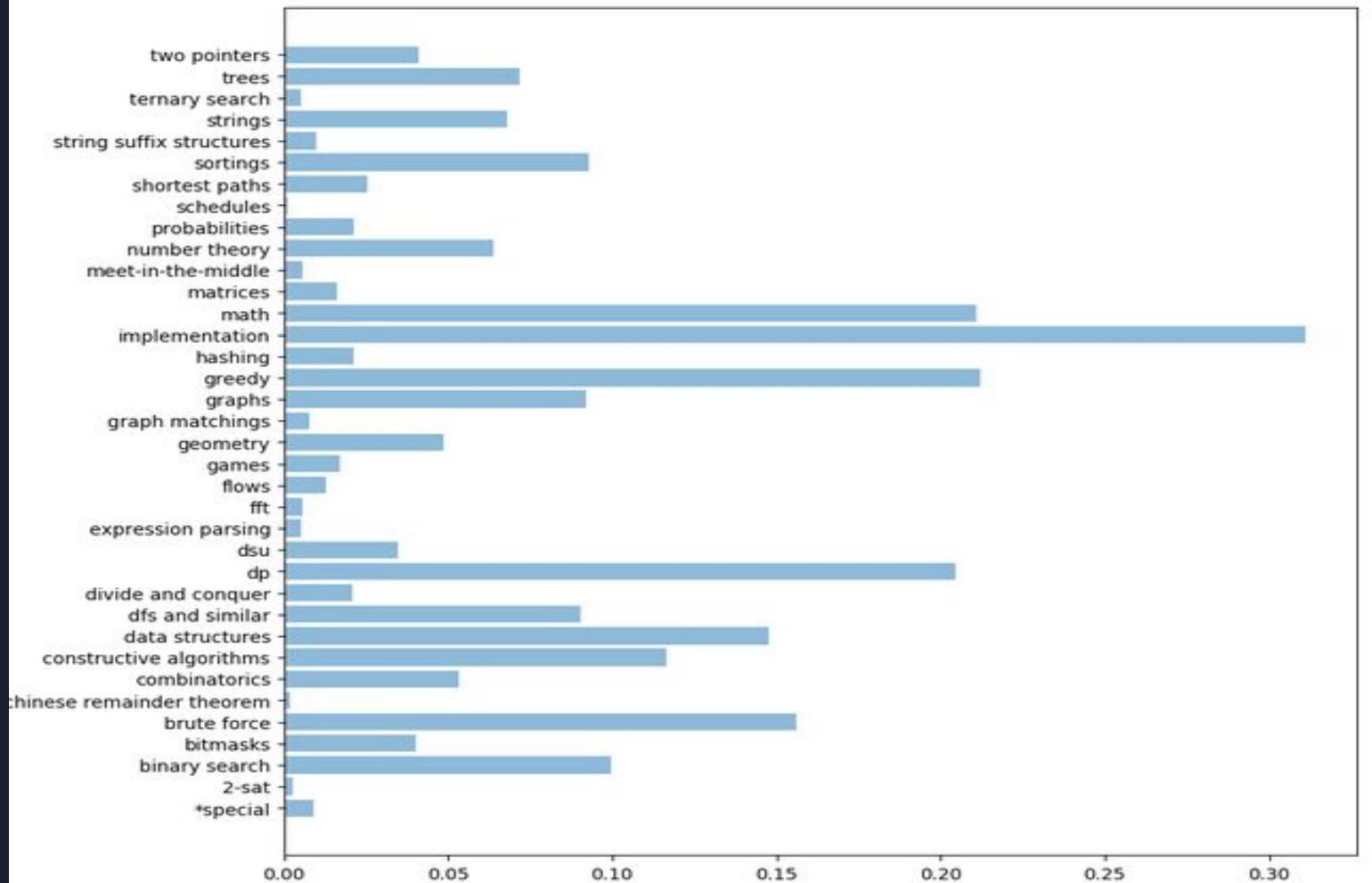
Editorial	Tags
http://disc	khadarbasha, 'simple-math', 'ad-hoc', nov12, 'cakewalk'
http://disc	['khadarbasha', 'simple-math', 'ad-hoc', 'nov12', 'cakewalk']
http://disc	['khadarbasha', 'simple-math', 'ad-hoc', 'nov12', 'cakewalk']
http://disc	['khadarbasha', 'simple-math', 'ad-hoc', 'nov12', 'cakewalk']
http://disc	['adhoc', berezin, 'easy', april14']

Tag List Used

```
global tags_list_codeforces , tags_list_codechef
```

```
tags_list_codeforces = ['dsu', 'trees', 'chinese remainder theorem', 'sortings', 'games', 'implementation', 'bitmask',  
                        '*special', 'hashing', 'geometry', 'two pointers', 'combinatorics', 'flows', 'strings',  
                        'probabilities', 'data structures', 'ternary search', 'greedy', 'math', 'matrices',  
                        'divide and conquer', 'dfs and similar', 'constructive algorithms', 'brute force', 'dp',  
                        '2-sat', 'graph matchings', 'binary search', 'number theory', 'graphs', 'fft', 'shortest paths',  
                        'schedules', 'meet-in-the-middle', 'string suffix structures', 'expression parsing']
```

```
tags_list_codechef = ['tree', 'binarysearch', 'combinatorial', 'gcd', 'dijkstra', 'memoization', 'bipartite', 'fibonacci',  
                     'strings', 'suffix', 'geometry', 'knapsack', 'sorting', 'recursion', 'pointers', 'maxflow', 'binary', 'fermat',  
                     'constructive', 'expo', 'graph', 'simulation', 'fft', 'algorithm', 'dfs', 'heap', 'bitmasking', 'hashing',  
                     'combinatorics', 'graphs', 'greedy', 'interactive', 'bfs', 'implementation', 'advanced', 'number', 'parity',  
                     'prime', 'dynamic', 'deque', 'sets', 'disjoint', 'bitwise', 'digraph', 'theory', 'backtracking', 'probability',  
                     'series', 'matrix', 'divide', 'kruskal', 'pattern', 'bruteforce', 'easy', 'hard', 'trees', 'maths', 'enumeration',  
                     'regex', 'tries', 'algebra', 'matching', 'multiset', 'euler', 'inversions', 'array', 'segment', 'permutation',  
                     'recurrence', 'dp', 'adhoc']
```



How labels look like



tags	
['algebra']	55
['bitwise']	91
['combinatorics', 'hard']	62
['combinatorics']	226
['constructive', 'gcd']	6
['constructive', 'greedy', 'implementation']	96
['disjoint', 'set']	3
['dp']	152
['game']	116
['graph', 'hard']	91
['graph']	95
['graphs']	31
['greedy']	1713
['hard', 'dp']	9
['hard', 'prime']	14
['hard']	145
['implementation']	201
['map']	3
['matching', 'graph']	7
['maths', 'matrix', 'expo']	244
['matrix', 'expo']	39
['series', 'algebra']	51
['tree']	1550



Proposed models

- ❏ Baseline models:
 - ❏ SVM
 - ❏ Logistic regression
 - ❏ LSTM
 - ❏ CNN
- ❏ BERT
- ❏ RoBERTa
- ❏ XLNET



Baseline models

- ❑ For Multilabel classification, we use One-vs-Rest scheme and use a Pipeline.
- ❑ Advantages:
 - ❑ Provides probabilities for each label.
 - ❑ It is more robust: the independent variables don't have to be normally distributed
- ❑ Disadvantages:
 - ❑ Linear model



Baseline models

Model	Precision	Recall
Logistic Regression	0.72	0.87
Linear SVM	0.76	0.82



Advanced models

As mentioned in the last evaluation we tried to implement BERT and Roberta this time as our model. Bert is transformer based model that has achieved state-of-the-art results on various NLP tasks. BERT is a bidirectional model , it uses much faster Attention-based approach. The model is also pre-trained on text corpus from wikipedia. This allows us to use a pre-trained BERT model by fine-tuning it for our multilabel classification task.



LRAP and Loss evaluations

Label ranking average precision (LRAP) averages over the samples, answers to the question: for each ground truth label, what fraction of higher-ranked labels were true labels? This performance measure will be higher if you are able to give better rank to labels associated with each sample.

This metric is used in multilabel ranking problem, where the goal is to give better rank to labels associated to each sample.

The obtained score is always strictly greater than 0 and the best value is 1.

The label ranking loss function computes ranking loss which averages over the samples the number of label pairs that are incorrectly ordered, i.e. true labels have a lower score than false labels



BERT

After preprocessing the data we fed the problem statement and its solutions to model along with labels after tuning the learning rate, epochs, optimiser we predicted probabilities for each label. We obtained LRAP as 0.91 and eval_loss as 0.06.



Roberta

RoBERTa builds on BERT's language masking strategy and modifies key hyperparameters in BERT, including removing BERT's next-sentence pretraining objective, and training with much larger mini-batches and learning rates. RoBERTa was also trained on an order of magnitude more data than BERT, for a longer amount of time. This allows RoBERTa representations to generalize even better to downstream tasks compared to BERT.

LRAP =0.92 eval_loss =0.04



Future Tasks

Fine tune advanced models to predict the test data.

Adjust parameters in all models so as to handle the unbalanced dataset.

Implementation of more basic as well as advanced models to check the best optimal one for this task.

Web app for user to fetch the questions and their respective solutions on input of corresponding label.



Managing Imbalanced Classes


As it is evident, the problem is of Multilabel classification.

The data we extracted had imbalance of classes, i.e. number of occurrence of some classes were more than the others.

Imbalanced classes put “accuracy” out of business. Standard accuracy no longer reliably measures performance.

Methods to deal with it:

- Up-sample Minority Class
- Down-sample Majority Class



Managing Imbalanced Classes Cont...

Up-sampling is the process of **randomly duplicating observations from the minority class** in order to reinforce its signal.

Down-sampling involves **randomly removing observations from the majority class** to prevent its signal from dominating the learning algorithm.

As we didn't have ton of data to work with, our best bet was up-sampling, which we used to remove class imbalance from the data.



LSTM

- ❑ An artificial recurrent neural network (RNN) architecture used in the field of deep learning.
- ❑ It has a “memory” which captures information about what has been calculated so far.
- ❑ Advantages :
 - ❑ It can process inputs of any length.
 - ❑ Even if the input size is larger, the model size does not increase.
- ❑ Disadvantages :
 - ❑ Due to its recurrent nature, the computation is slow.
 - ❑ Training of LSTM models can be difficult.



LSTM

A threshold matrix has been defined, with values in range 0.1 to 0.9. Then, we run a loop over the predicted output and compare it with the threshold value and choose tags only if the corresponding value of a tag is more than the threshold value.

Precision : 0.3091
Recall : 0.3638
F1 - score : 0.3342

```
For threshold: 0.1
Micro-average quality numbers
Precision: 0.2298, Recall: 0.4838, F1-measure: 0.3116
For threshold: 0.2
Micro-average quality numbers
Precision: 0.3091, Recall: 0.3638, F1-measure: 0.3342
For threshold: 0.3
Micro-average quality numbers
Precision: 0.3663, Recall: 0.2902, F1-measure: 0.3238
For threshold: 0.4
Micro-average quality numbers
Precision: 0.4177, Recall: 0.2345, F1-measure: 0.3004
For threshold: 0.5
Micro-average quality numbers
Precision: 0.4738, Recall: 0.1933, F1-measure: 0.2746
For threshold: 0.6
Micro-average quality numbers
Precision: 0.5327, Recall: 0.1572, F1-measure: 0.2427
For threshold: 0.7
Micro-average quality numbers
Precision: 0.5785, Recall: 0.1246, F1-measure: 0.2050
For threshold: 0.8
Micro-average quality numbers
Precision: 0.6372, Recall: 0.0927, F1-measure: 0.1618
For threshold: 0.9
Micro-average quality numbers
Precision: 0.6852, Recall: 0.0601, F1-measure: 0.1105
```



CNN

- ❑ **CNN** or **ConvNet** is a class of deep neural networks.
- ❑ **Advantages :**
 - ❑ CNNs eliminate the need for manual feature extraction.
 - ❑ CNNs can be retrained for new recognition tasks, enabling us to build on pre-existing networks.
- ❑ **Disadvantages :**
 - ❑ They need a lot of training data.
 - ❑ If GPU is not good, they are quite slow to train (for complex tasks).



CNN

A threshold matrix has been defined, with values in range 0.1 to 0.9. Then, we run a loop over the predicted output and compare it with the threshold value and choose tags only if the corresponding value of a tag is more than the threshold value.

Precision : 0.3369
Recall : 0.3674
F1 - score : 0.3515

```
For threshold: 0.1
Micro-average quality numbers
Precision: 0.2299, Recall: 0.5640, F1-measure: 0.3266
For threshold: 0.2
Micro-average quality numbers
Precision: 0.3369, Recall: 0.3674, F1-measure: 0.3515
For threshold: 0.3
Micro-average quality numbers
Precision: 0.4548, Recall: 0.2483, F1-measure: 0.3212
For threshold: 0.4
Micro-average quality numbers
Precision: 0.5458, Recall: 0.1745, F1-measure: 0.2644
For threshold: 0.5
Micro-average quality numbers
Precision: 0.6115, Recall: 0.1343, F1-measure: 0.2203
For threshold: 0.6
Micro-average quality numbers
Precision: 0.6531, Recall: 0.0927, F1-measure: 0.1623
For threshold: 0.7
Micro-average quality numbers
Precision: 0.7085, Recall: 0.0488, F1-measure: 0.0913
For threshold: 0.8
Micro-average quality numbers
Precision: 0.7125, Recall: 0.0126, F1-measure: 0.0248
For threshold: 0.9
Micro-average quality numbers
Precision: 0.0000, Recall: 0.0000, F1-measure: 0.0000
```




XLNET

It is a large bidirectional transformer that uses improved training methodology, larger data and more computational power to achieve better than BERT prediction.

It uses permutation language modeling, where all tokens are predicted but in random order. This is in contrast to BERT's masked language model where only the masked (15%) tokens are predicted.. This helps the model to learn bidirectional relationships and therefore better handles dependencies and relations between words.

XLNet was trained with over 130 GB of textual data and 512 TPU chips running for 2.5 days, both of which are much larger than BERT hence it takes larger training time but outperforms BERT in many tasks.



Experimentation with models in their learning rate, epochs, max sequence length etc we came to following conclusion in advance models. As it takes lot of time to train these advanced models we trained only on segment of data taking only 5000 examples with balanced tags. The best results obtained so far are.

BERT

Learning rate= $5e-5$

Training epochs = 3

LRAP = 0.98

Eval-Loss= 0.013



ROBERTA

Learning rate= $5e-5$

Training epochs = 3

LRAP = 0.99

Eval-Loss= 0.008




XLNET

Learning rate= $5e-5$

Training epochs = 1

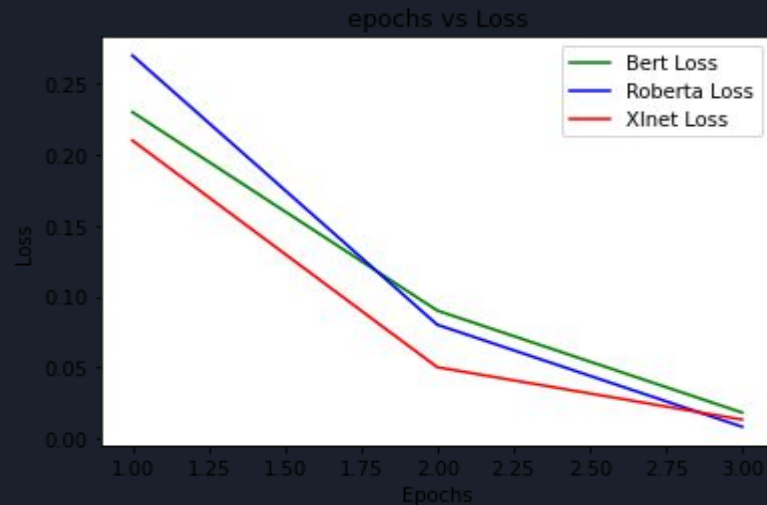
LRAP = 0.99

Eval-Loss= 0.018



We can see difference between three models as xlnet is more resource intensive and takes longer time to train as compared to the other two.

There is not much difference in performance however Roberta performs best.



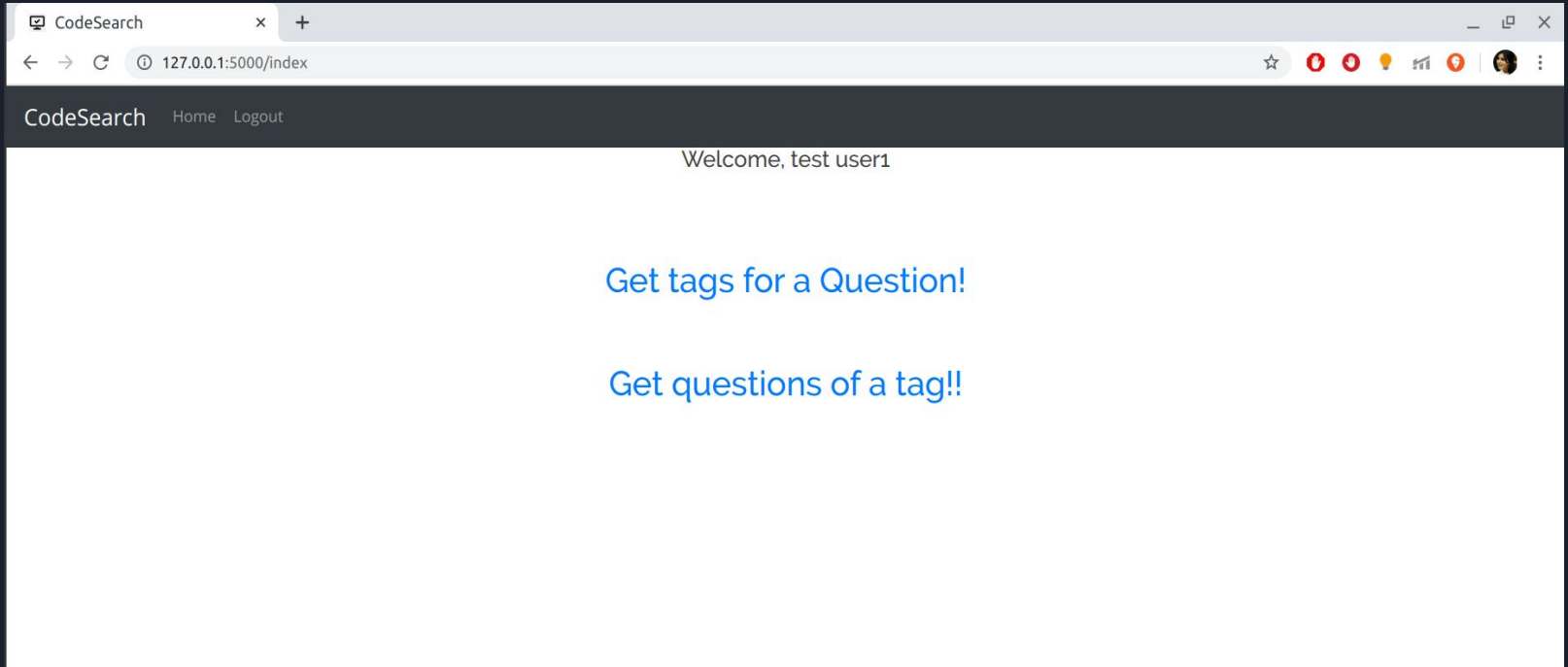


Web Application

Web application provides the following functionalities:

- ❑ To obtain tags for the question of interest.
- ❑ To provide questions belonging to a paradigm from different platforms at one place.
- ❑ Login
- ❑ Register

Web Application - Home



Web Application

CodeSearch

127.0.0.1:5000/find_questiontag

☆ 🔴 🔴 💡 📶 🔴 👤 ⋮

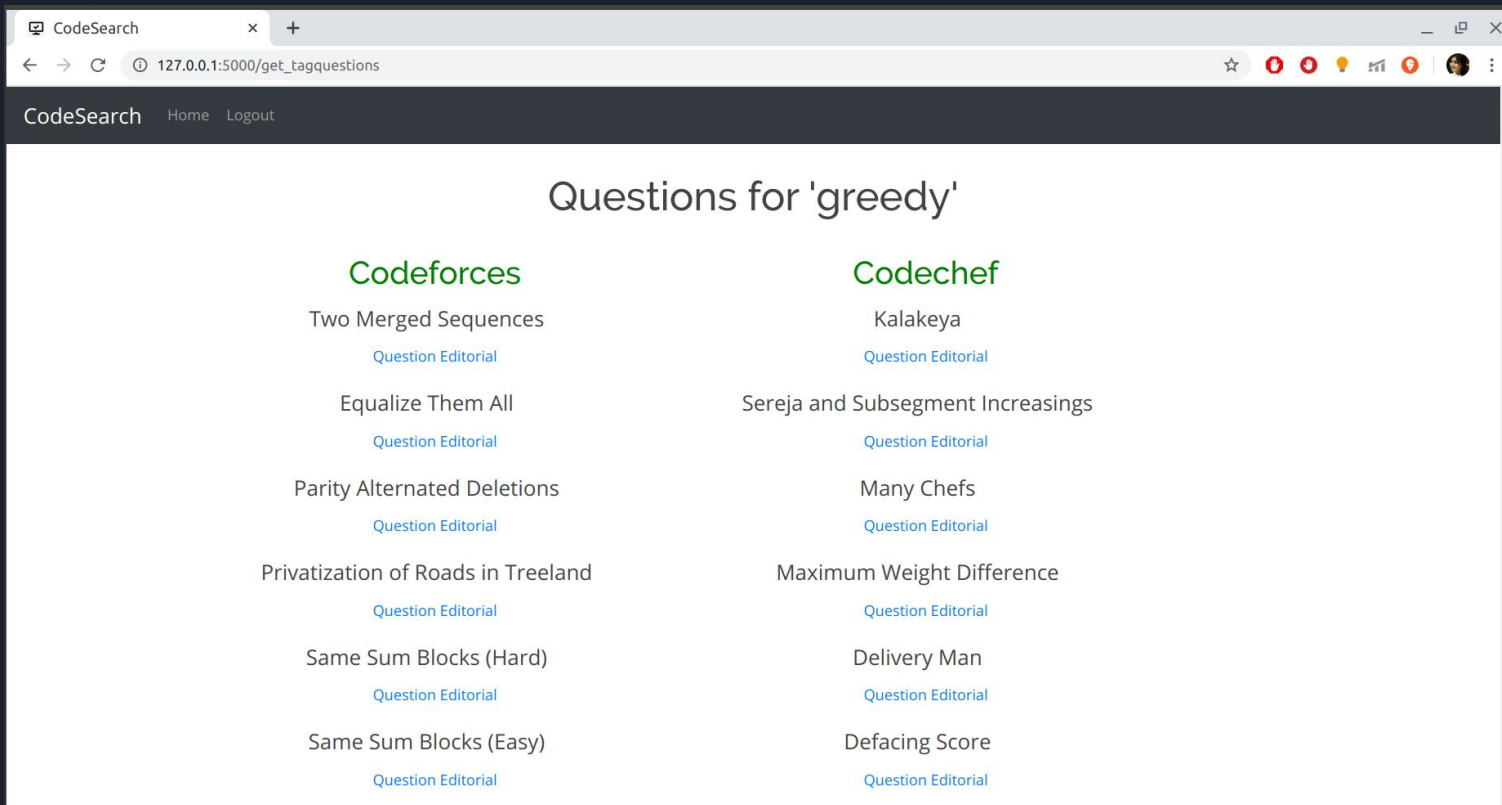
CodeSearch Home Logout

Tag(s) for given question

constructive, greedy, implementation

All submissions for this problem are available. The Kalakeyas were a powerful, ferocious and cruel clan of Danavas. They were known to be really strong and they did not have any war strategy. They would just attack the enemy randomly and overpower them with sheer number of soldiers. However, we all know that Baahubali and Bhallaladeva defeated the Kalakeyas by following the Thrishul strategy, and successfully defended their kingdom Maahishmati. We also know that Baahubali was very smart, and the truth is that he predicted how the Kalakeyas would attack and devised a counter strategy for the same, the night before the war. This is what he found: The Kalakeyas had N forts, numbered 1 to N and Baahubali had N soldiers, numbered 1 to N . Baahubali discovered that he can permute his soldiers in any way to get a permutation of 1 to $N \Rightarrow P_1, P_2, \dots, P_N$. He would then send his soldiers to attack the forts in the following way: soldier P_1 attacks fort 1, soldier P_2 attacks fort 2, ..., soldier P_N attacks fort N . It is easy to note that each soldier attacks exactly one fort and no two soldiers attack the same fort. Baahubali also got to know about a secret key of the Kalakeyas, which is an integer K . A soldier X can destroy a fort Y , iff $\text{abs}(X - Y) \geq K$. For more details on the $\text{abs}()$ function, check here. Your task is to determine whether Baahubali's soldiers can be permuted in some way, such that all forts can be destroyed. In other words, for a permutation P_1, P_2, \dots, P_N , Baahubali's soldiers can destroy all the forts iff $\text{abs}(P_i - i) \geq K$, for all $1 \leq i \leq N$. If this is possible, you are also required to output the lexicographically smallest such permutation. If it is not possible, output -1. Note: A permutation A_1, A_2, \dots, A_N is said to be lexicographically smaller than a permutation B_1, B_2, \dots, B_N , if and only if at the first i where A_i and B_i differ, A_i comes before B_i . You can refer here for a more detailed definition of lexicographic ordering. Input The first line of input consists of a single integer T denoting the number of test cases. Each of the following T lines contain two space separated integers N and K denoting the values mentioned in the statement above. Output For each test case, output a single line containing N space separated integers (which should be a permutation of $[1..N]$), if Baahubali's soldiers can break all the forts. If it is not possible to break all the forts, output "-1" (quotes for clarity). Constraints $1 \leq T \leq 1000$ $1 \leq N \leq 105$ $0 \leq K \leq N$ The sum of N over all test cases in a single test file will not exceed 105 Example Input: 3 2 3 0 3 1 Output: -1 1 2 3 2 3 1 Explanation For the first test case, $N = 2$ and $K = 2$. It is impossible to permute $[1, 2]$ in any way such that $\text{abs}(P[1]-1) \geq 2$ and $\text{abs}(P[2]-2) \geq 2$. Hence, output is -1. For the second test case, $N = 3$ and $K = 0$. We can just set $P[i] = i$, and hence the answer is 1 2 3 For the third case, the valid permutations are $[2, 3, 1]$ and $[3, 1, 2]$. The answer is $[2, 3, 1]$ since it is lexicographically smaller than $[3, 1, 2]$.

Web Application





Github Link

<https://github.com/sai-sukrutha/Coding-Questions-Classifier>



Thank You