Problemas de Programação - Projeto 1

Recuperação da Informação - IF962

Douglas Soares Lins **(dsl)**
Jônatas de Oliveira Clementino **(joc)**
Valdemiro Rosa Vieira Santos **(vrvs)**

CIn - UFPE
Recife - 10 de Maio de 2018

Domínios

- https://wcipeg.com
- http://www.codeforces.com
- https://a2oj.com
- https://www.codechef.com (problema)
- http://www.spoj.com
- https://dmoj.ca (OBS)
- http://acm.timus.ru
- https://www.urionlinejudge.com.br (problema)
- https://leetcode.com (problema)

1.

Crawler

Valdemiro Vieira - vrvs

Pontos Importantes

- **Respeitar o robots.txt**
  - Robotparser (problema)
  - Reppy
- **Baixar apenas páginas html**
  - text/html
- **Evitar sobrecarregar páginas**
  - PhantomJS lento
- **Manter no domínio**
  - Não visitar páginas externas

Problemas

- **Algumas páginas tinham conteúdo importante no JS**
  - Requests não carregava
  - Solução: Selenium + PhantomJS
- **Leet bem instável**
  - Não determinístico. Às vezes carregava tudo. Achamos melhor não incluí-la
- **Extremamente lento**
  - Solução: limitar crescimento da fila até certo valor
- **Manter no domínio**
  - Não visitar páginas externas

Estratégias de Visitação

- **Breadth-First Search (BFS)**
  - Estratégia básica
- **Heurística**
  - Analisar endereços URL de cada domínio
  - Usando fila de prioridade
  - Maior prioridade para áreas do site que levam aos problemas
  - Evitar crescimento (ser mais eficiente)
- **Heurística 2.0**
  - Aumentar um pouco o crescimento
  - Só colocar na fila de prioridade páginas com certa relevância

Resultados - Harvest Ratio

Domínio	BFS	Heurística	Heurística 2.0	Páginas
A2OJ	0%	36,6%	96,3%	273
Codeforces	10,4%	98,6%	98,4%	1000
Spoj	29,9%	52,8%	94,8%	1000
Timus	75,5%	50,9%	98,4%	1000
Wcipeg	25,2%	32,6%	89,8%	1000

Perguntas?

2.
Classificador

Jônatas Oliveira - joc

Processo de Classificação

- Foram selecionados em média 24-26 documentos de cada domínio, para serem instâncias do classificador. Sendo metade documentos positivos e outra metade negativos.
- Teste com 6 diferentes tipos de classificadores, encontrados na lib Scikit Learn de python.
- O método de *bag of words* foi utilizado para selecionar *features* a partir desse conjunto.
- Uma observação é que no classificador Naive Bayes foi utilizado Multinomial Naive Bayes.

Problemas

- **Algumas páginas tinham conteúdo importante no JS**
  - Requests não carregava
  - Solução: Selenium + PhantomJS
- **Página vinha com info de estruturas e scripts (mesmo só recebendo informações textuais)**
  - Verificado que as informações textuais das estruturas auxiliam na classificação
  - Porém scripts (códigos js) foram retirados
- **Classificação somente por frequência não resultava em grande precisão**
  - Solução: treinar uma árvore de decisão com o conjunto de palavras dos documentos, e escolher features a partir do ganho de informação.

Tempo de Treinamento

Classificadores	Tempo em Segundos
Naive Bayes	0.195
Árvore de Decisão	0.198
MLP	0.268
SVM	0.388
KNN	0
Logistic Regression	1.020

Resultados

	Recall	Precision	F-Measure	Accuracy
Naive Bayes	100%	89,67%	94,55%	94,30%
Árvore de decisão	100%	100%	100%	100%
MLP	100%	100%	100%	100%
SVM	97,64%	98,93%	98,28%	98,30%
KNN	100%	96,54%	98,24%	98,19%
Logistic Regression	100%	98,96%	99,48%	99,47%

3.
Extrator

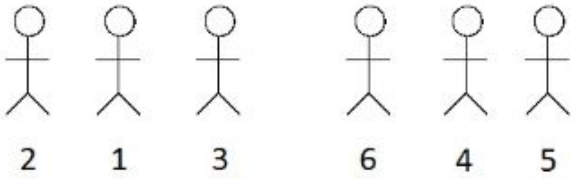
Douglas Soares - dsl

Problemas

- **Algumas páginas tinham conteúdo importante no JS;**
  - Requests não carregava
  - Solução: Selenium + PhantomJS
- **No extrator genérico, em alguns casos vinha código JS do site e CSS;**
- **Estruturas de páginas do mesmo domínio diferentes.**

G: Office Mates

Dr. Baws has an interesting problem. His N graduate students, while friendly with some select people, are generally not friendly with each other. No graduate student is willing to sit beside a person they aren't friends with.



The desks are up against the wall, in a single line, so it's possible that Dr. Baws will have to leave some desks empty. He does know which students are friends, and fortunately the list is not so long: it turns out that for any subset of K graduate students, there are at most $K - 1$ pairs of friends. Dr. Baws would like you to minimize the total number of desks required. What is this minimum number?

Point Value: 20
Time Limit: 2.00s
Memory Limit: 64M
Added: Jul 08, 2014
Author: SourSpinach

Input

The input begins with an integer $T \leq 50$, the number of test cases. Each test case begins with two integers on their own line: $N \leq 100000$, the number of graduate students (who are indexed by the integers 1 through N), and M, the number of friendships among the students. Following this are M lines, each containing two integers i and j separated by a single space. Two integers i and j represent a mutual friendship between students i and j.

The total size of the input file does not exceed 2 MB.

Output

For each test case output a single number: the minimum number of desks Dr. Baws requires to seat the students.

Sample Input

```
1
6 5
1 2
1 3
1 4
4 5
4 6
```

Sample Output

```
7
```

JULKA - Julka

#big-numbers

Julka surprised her teacher at preschool by solving the following riddle:

Klaudia and Natalia have 10 apples together, but Klaudia has two apples more than Natalia. How many apples does each of he girls have?

Julka said without thinking: Klaudia has 6 apples and Natalia 4 apples. The teacher tried to check if Julka's answer wasn't accidental and repeated the riddle every time increasing the numbers. Every time Julka answered correctly. The surprised teacher wanted to continue questioning Julka, but with big numbers she could't solve the riddle fast enough herself. Help the teacher and write a program which will give her the right answers.

Task

Write a program which

- reads from standard input the number of apples the girls have together and how many more apples Klaudia has,
- counts the number of apples belonging to Klaudia and the number of apples belonging to Natalia,
- writes the outcome to standard output

Input

Ten test cases (given one under another, you have to process all!). Every test case consists of two lines. The first line says how many apples both girls have together. The second line says how many more apples Klaudia has. Both numbers are positive integers. It is known that both girls have no more than 10^{100} (1 and 100 zeros) apples together. As you can see apples can be very small.

Output

For every test case your program should output two lines. The first line should contain the number of apples belonging to Klaudia. The second line should contain the number of apples belonging to Natalia.

Example

```
Input:
10
2
[and 9 test cases more]

Output:
6
4
[and 9 test cases more]
```

Added by:	Adam Dzedzej
Date:	2004-06-08
Time limit:	2s
Source limit:	50000B
Memory limit:	1536MB
Cluster:	Cube (Intel G860)
Languages:	All except: NODEJS PERL6 VB.NET
Resource:	Internet Contest **Pogromcy Algorytmow** (Algorithm Tamers) Round II, 2003

Resultados

	Recall	Precision	F-Measure
A2oj	100%	100%	100%
CodeForces	100%	100%	100%
DMOJ	100%	100%	100%
LeetCode	100%	100%	100%
SPOJ	98,5%	98,5%	98,5%
Timus	100%	100%	100%
WCIPEG	89,8%	89,8%	89,8%

Extratores Genéricos

Baseado em Tags

Utilizado as tags mais recorrentes dos corpos das páginas onde a informação que importa está.

["p", "h2", "h3", "h4", "pre", "br"]

Resultados

	Recall	Precision	F-Measure
A2oj	50,7%	54,2%	52,4
CodeForces	1,3%	5%	2,1%
DMOJ	70,8%	98%	82,2%
LeetCode	100%	100%	100%
SPOJ	47%	65,3%	54,7%
Timus	13,5%	50%	21,2%
WCIPEG	46,3%	74,4%	57,1%

Baseado em Keywords

Utilizado keywords mais relevantes para a extração.

Input
Output
Example
Sample
Note

Constraint
Follow up
Time Limit
Memory Limit

Baseado em Keywords

Todos vem com o corpo que contém as informações desejadas para extrair, mas alguns vêm com informações a mais, como comentários e outros com as informações tudo colado.

```
Description:
B. Fractaltime limit per test2 secondsmemory limit per test64 megabytesinputinput.txtoutputoutput.txtEver since Kalevitch, a famous Berland abstractionist, heard of fractals, he made them the main
 topic of his canvases. Every morning the artist takes a piece of graph paper and starts with making a model of his future canvas. He takes a square as big as n × n squares and paints some of them
 black. Then he takes a clean square piece of paper and paints the fractal using the following algorithm: Step 1. The paper is divided into n2 identical squares and some of them are painted black
according to the model.Step 2. Every square that remains white is divided into n2 smaller squares and some of them are painted black according to the model.Every following step repeats step 2.  Un
fortunately, this tiresome work demands too much time from the painting genius. Kalevitch has been dreaming of making the process automatic to move to making 3D or even 4D fractals.InputThe first
line contains integers n and k (2 ≤ n ≤ 3, 1 ≤ k ≤ 5), where k is the amount of steps of the algorithm. Each of the following n lines contains n symbols that determine the model. Symbol «.» stands
 for a white square, whereas «*» stands for a black one. It is guaranteed that the model has at least one white square. OutputOutput a matrix nk × nk which is what a picture should look like after
 k steps of the algorithm.ExamplesInputCopy2 3.*..OutputCopy.*******..******.*.*****....****.***.***..**..**.*.*.*.*........InputCopy3 2.*.***.*.OutputCopy.*.***.*.*********.*.***.*.**************
*************.*.***.*.*********.*.***.*.
```

B. Fractal

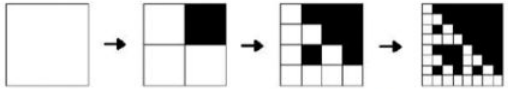
time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: input.txt
output: output.txt

Ever since Kalevitch, a famous Berland abstractionist, heard of fractals, he made them the main topic of his canvases. Every morning the artist takes a piece of graph paper and starts with making a model of his future canvas. He takes a square as big as $n \times n$ squares and paints some of them black. Then he takes a clean square piece of paper and paints the fractal using the following algorithm:

Step 1. The paper is divided into n^2 identical squares and some of them are painted black according to the model.

Step 2. Every square that remains white is divided into n^2 smaller squares and some of them are painted black according to the model.

Every following step repeats step 2.



Unfortunately, this tiresome work demands too much time from the painting genius. Kalevitch has been dreaming of making the process automatic to move to making 3D or even 4D fractals.

Input

The first line contains integers n and k ($2 \le n \le 3$, $1 \le k \le 5$), where k is the amount of steps of the algorithm. Each of the following n lines contains n symbols that determine the model. Symbol «.» stands for a white square, whereas «*» stands for a black one. It is guaranteed that the model has at least one white square.

Output

Output a matrix $n^k \times n^k$ which is what a picture should look like after k steps of the algorithm.

Examples

input Copy

```
2 3
.*
..
```

output Copy

```
.*******
..******
.*.*****
....****
.***.***
..**..**
.*.*.*.*
........
```

input Copy

```
3 2
.*.
***
.*.
```

output Copy

```
.*.***.*.
*********
.*.***.*.
*********
*********
*********
.*.***.*.
*********
.*.***.*.
```

Description:
B. Fractaltime limit per test2 secondsmemory limit per test64 megabytesinputinput.txtoutputoutput.txtEver since Kalevitch, a famous Berland abstractionist, heard of fractals, he made them the main topic of his canvases. Every morning the artist takes a piece of graph paper and starts with making a model of his future canvas. He takes a square as big as n × n squares and paints some of them black. Then he takes a clean square piece of paper and paints the fractal using the following algorithm: Step 1. The paper is divided into n2 identical squares and some of them are painted black according to the model.Step 2. Every square that remains white is divided into n2 smaller squares and some of them are painted black according to the model.Every following step repeats step 2. Unfortunately, this tiresome work demands too much time from the painting genius. Kalevitch has been dreaming of making the process automatic to move to making 3D or even 4D fractals.InputThe first line contains integers n and k (2 ≤ n ≤ 3, 1 ≤ k ≤ 5), where k is the amount of steps of the algorithm. Each of the following n lines contains n symbols that determine the model. Symbol «.» stands for a white square, whereas «*» stands for a black one. It is guaranteed that the model has at least one white square. OutputOutput a matrix nk × nk which is what a picture should look like after k steps of the algorithm.ExamplesInputCopy2 3.*..OutputCopy.*******..******.*.*****....****.***.***..**..**.*.*.*.*........InputCopy3 2.*.***.*.OutputCopy.*.***.*.*********.*.***.*.*********.*.***.*.*********.*.***.*.

Resultados

	Recall	Precision	F-Measure
A2oj	52,3%	52,3%	52,3%
CodeForces	0%	0%	0%
DMOJ	68%	87,5%	76,5%
LeetCode	42,1%	42,1%	42,1%
SPOJ	32,3%	34,3%	33,3%
Timus	13,5%	50%	21,2%
WCIPEG	49,2%	65,3%	56,1%

Comparação dos extratores genéricos

	G. Tags	G. Keywords
Recall	32,9%	25,7%
Precision	44,7%	33,1%
F-Measure	37%	28,1%

```
def extractor(page, domain, crawlerType, fileName):

    os.makedirs('Docs/Jsons/' + crawlerType + '/Specific/' + domain, exist_ok=True)
    os.makedirs('Docs/Jsons/' + crawlerType + '/General_1/' + domain, exist_ok=True)
    os.makedirs('Docs/Jsons/' + crawlerType + '/General_2/' + domain, exist_ok=True)

    if domain == "A2oj":
        a2oj.a2oj(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Codeforces":
        codeforces.codeforces(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Dmoj":
        dmoj.dmoj(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Leetcode":
        leetcode.leetcode(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Spoj":
        spoj.spoj(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Timus":
        timus.timus(page, crawlerType, "Specific", domain, fileName)
    elif domain == "Wcipeg":
        wcipeg.wcipeg(page, crawlerType, "Specific",domain,  fileName)

    g1.genericExtractor(page, crawlerType, "General_1", domain, fileName)
    g2.genericExtractor2(page, crawlerType, "General_2", domain, fileName)
```

Json


```
{
  "Description": "\nProblem Statement:\n\n\nOne day, your friend Ahmed wanted to challenge you with the hardest challenge ever, watching snails for so long time!\n\nAhmed has n s
  "Notes": "\nNotes:\n\n\nHere is an explanation of the second test case:\n\nmax_meeting = 2\n\n\nmin_meeting = 0\n\n\n\n\n",
  "Example": "\nSample Input:\n\n3\n1 5\n2 1\n3 1\n\n\nSample Output:\n\n0 0\n0 2\n0 4\n\n",
  "Time Limit": "3 seconds",
  "Input Description": "\nInput Format:\n\n\nYour program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cas
  "Output Description": "\nOutput Format:\n\n\nFor each test case print in a single line two numbers min_meetings and max_meetings separated by a single space.\n\n\n",
  "Title": "134. Slow Snails"
}{
  "Description": "\nProblem Statement:\n\n\nWhen we were kids, we used to play with some stickers where these stickers contain some (but not necessarily all) lower case English a
  "Input Description": "\nInput Format:\n\n\nYour program will be tested on one or more test cases. The first line of the input will be a single integer T , the number of test ca
  "Example": "\nSample Input:\n\n4\naa??bb\naaccbb\n?a?\na??a\n\n\nSample Output:\n\n3\n1\n1\n1\n\n",
  "Title": "2. The Alphabet Sticker",
  "Output Description": "\nOutput Format:\n\n\nFor each test case, print a single line which contains a single integer representing the number of possible original configurations
  "Time Limit": "3 seconds",
  "Notes": ""
}
```

Obrigado!

Douglas Soares Lins **(dsl)**
Jônatas de Oliveira Clementino **(joc)**
Valdemiro Rosa Vieira Santos **(vrvs)**