

Tempo Limite: 0,5 s (C/C++)

Algoritmos de classificação geralmente permitem a definição de parâmetros, conhecidos como hiperparâmetros, no processo de aprendizagem para criar um modelo que poderá levar a um certo desempenho preditivo em problemas de aprendizagem de máquina. Um novo conjunto de classificadores a serem testados para dados de um problema de seu interesse possui apenas 2 parâmetros X e Y que podem ser definidos usando valores inteiros maiores que zero. Para otimizar esses parâmetros em um processo de tuning, você pretende testar uma sequência de possíveis valores para esses parâmetros, pois cada par de valores leva a um determinado resultado preditivo, que você pretende, posteriormente, comparar com todos resultados obtidos de forma a escolher valores para esses parâmetros de cada algoritmo de classificação que resultem em um bom modelo. Porém, cada algoritmo possui diferentes regras para a escolha de valores desses parâmetros. Os valores que podem ser utilizados nos parâmetros podem ser visualizados como uma grade bidimensional, em que cada posição (x,y) da grade representa um par de possíveis valores para os parâmetros X e Y . Para posições fora da grade, ou seja, combinações não viáveis de valores, o algoritmo não funcionará. Portanto, de posse do formato da grade de possíveis valores de parâmetros de um dado algoritmo, a sua tarefa será a de gerar várias possíveis combinações de valores a serem testados para que você consiga determinar posteriormente um par de valores que obtém um desempenho de predição otimizado. A figura abaixo, mostra uma grade em que cada posição (x,y) representa um possível par de valores x e y para os parâmetros X e Y , respectivamente. O asterisco na primeira posição representa o primeiro par a ser testado $(1,1)$.

[illegible]

Para otimizar o tuning dos parâmetros, você pensou em testar uma nova estratégia que segue uma dada sequência de pares de valores que simula o movimento de um cavalo do jogo de xadrez, pois em alguns casos é interessante pular várias posições a partir da posição atual para poder chegar a um resultado ótimo mais rapidamente. Ou seja, para um certo par de valores (x,y) testado, a estratégia escolhe como próximo par de valores algum dos seguintes pares de valores: $(x-2,y-1)$, $(x-1,y-2)$, $(x+1,y-2)$, $(x+2,y-1)$, $(x-2,y+1)$, $(x-1,y+2)$, $(x+1,y+2)$ e $(x+2,y+1)$, isso se o próximo par de valores também for um par de valores válidos para o algoritmo, ou seja, uma posição que pertence à grade de possíveis valores.

Como cada par de valores só precisa ser testado uma vez, é importante que a sequência de valores não repita nenhum par durante o processo de tuning. Por outro lado, é interessante que seja possível testar o maior número de possíveis valores para se aproximar do melhor resultado preditivo possível para o dado algoritmo. Você deve implementar um programa que determina uma solução com o número máximo de pares de valores viáveis que podem ser testados pela estratégia do cavalo em sequência a partir do par de valores iniciais (1,1) e uma dada grade de valores permitidos pelo classificador.

Entrada

A entrada contém uma grade para um dado algoritmo de classificação. A primeira linha representa a quantidade de linhas N ($1 \leq N \leq 10$) da grade dos parâmetros. Em cada uma das próximas N linhas, será fornecido um par de inteiros a_i e b_i ($0 \leq a_i < 10$, $0 < b_i \leq 10$) referentes a uma dada linha i da grade, indicando que na linha i as a_i primeiras colunas não existem e, em seguida, existem b_i colunas. Cada grade terá, no máximo, 35 posições.

Saída

Para cada entrada, você deverá imprimir a quantidade de pares de valores que não seriam testados na solução ótima encontrada, como especificado no problema.

Exemplos

Entrada	Saída
6	3
0 6	
0 4	
0 4	
2 3	
2 3	
2 5	

Entrada	Saída
2	0
0 1	
2 1	

Entrada	Saída
3	1
0 3	
0 3	
0 3	