Departamento de Computação

Programação Competitiva unesp

1º Contest

Laboratório de Programação Competitiva I

Pedro Henrique Paiola

Rene Pegoraro

Wilson M Yonezawa

C – Beat the Spread! (UVA – 10812)

- **Problema:** calcular, se possível, o placar de dois times a partir da soma **s** das pontuações e da diferença absoluta **d**
- É garantido que os valores de entrada **s** e **d** são inteiros não negativos

C – Beat the Spread! (UVA – 10812)

- Considerando **x** como sendo a maior pontuação, e **y** a menor, podemos dizer que:

$$\begin{cases} s = x + y \\ d = x - y \end{cases}$$

- Resolvendo este sistema obtemos que:

C – Beat the Spread! (UVA – 10812)

$$x = \frac{s+d}{2} \qquad y = \frac{s-d}{2}$$

- Analisando estas equações, e sabendo que a pontuação de cada time é sempre um inteiro positivo, podemos dizer que só é possível calcular **x** e **y** quando:
  - $s + d$ e $s - d$ são números pares
  - $s - d \geq 0$ ou, de forma equivalente, $s \geq d$

D – Peter's Smokes (UVA – 10346)

- **Problema:** determinar quantos cigarros Peter pode possuir, sendo que ele começa com **n** cigarros e a cada **k** bitucas ele consegue enrolar um novo cigarro.

D – Peter’s Smokes (UVA – 10346)

- Este exercício pode ser resolvido apenas simulando o problema descrito.

```
soma = n;
bitucas = n;
while(bitucas >= k)
{
     n = bitucas/k;
     soma += n;
     bitucas = bitucas % k + n;
}
cout << soma << endl;
```

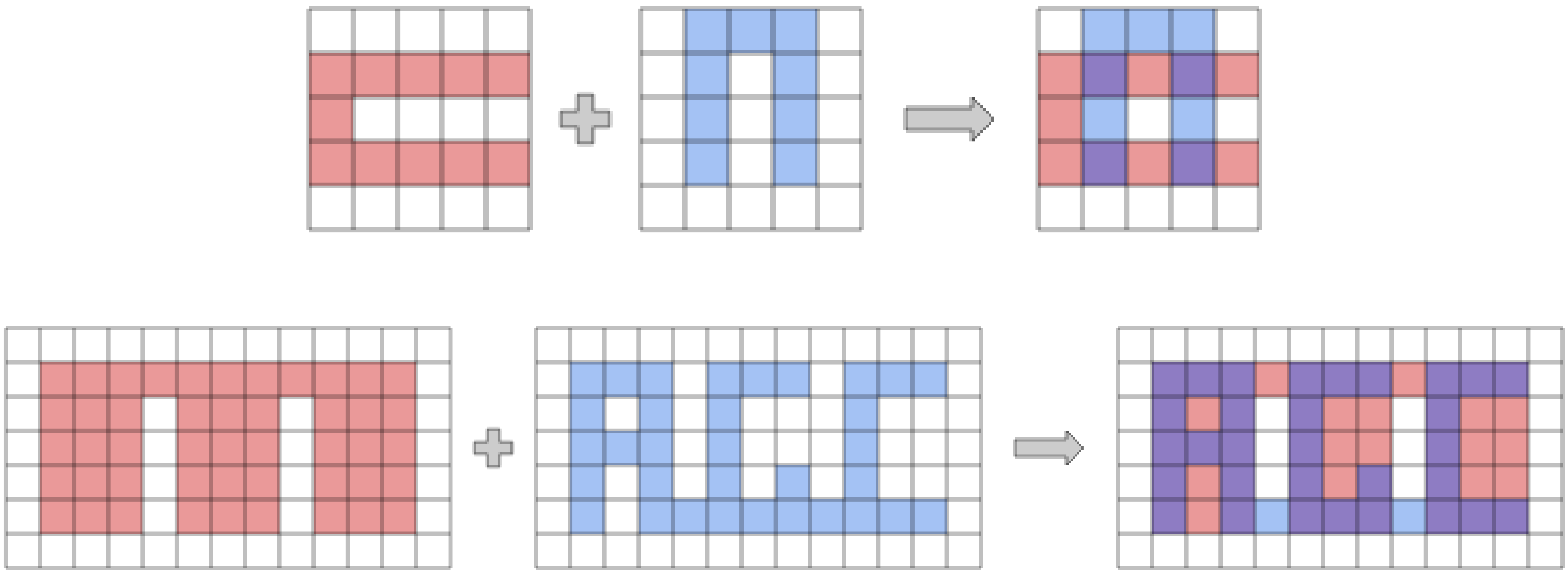
F – AND Grid (AtCoder – agc004_c)

- **Descrição:** uma determinada matriz possui células que podem ser coloridas de azul, vermelho ou por ambas as cores (a sobreposição torna a célula roxa), com as seguintes condições
  - Todas as células vermelhas devem fazer parte de uma mesma região conexa (4-*connected*)
  - Todas as células azuis devem fazer parte de uma mesma região conexa
  - Não deve haver nenhuma célula roxa nas bordas da matriz

F – AND Grid (AtCoder – agc004_c)

- **Problema:** Dada as posições das células roxas, determinar uma possível solução com uma região azul e uma região vermelha.
  - OBS: todas as células roxas são informadas pela entrada, ou seja, soluções que possuam mais intersecções entre as regiões azul e vermelha do que as informadas são inválidas

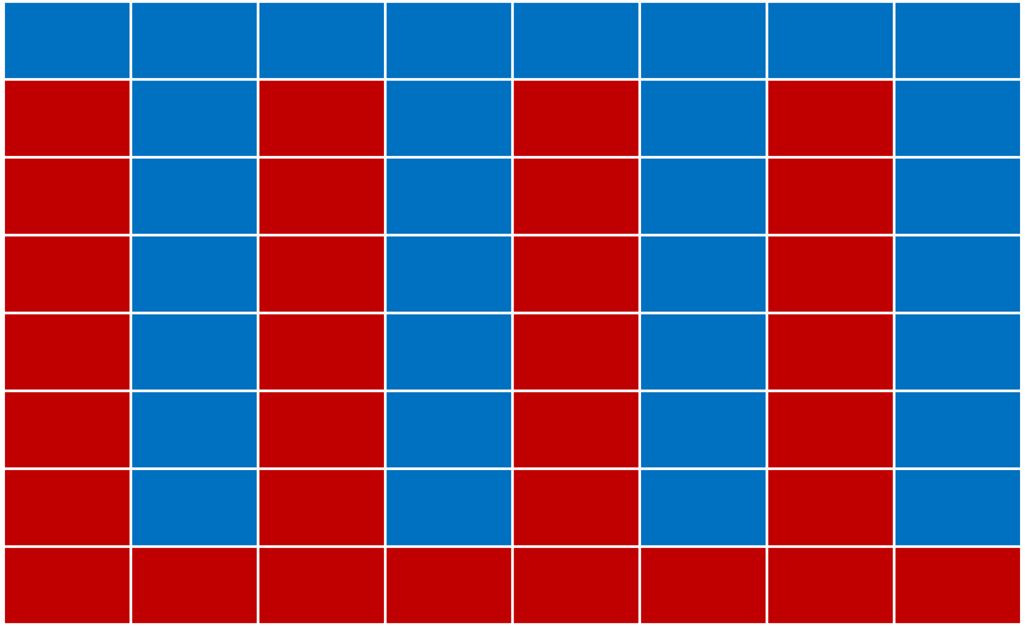
F - AND Grid (AtCoder - agc004_c)



F – AND Grid (AtCoder – agc004_c)

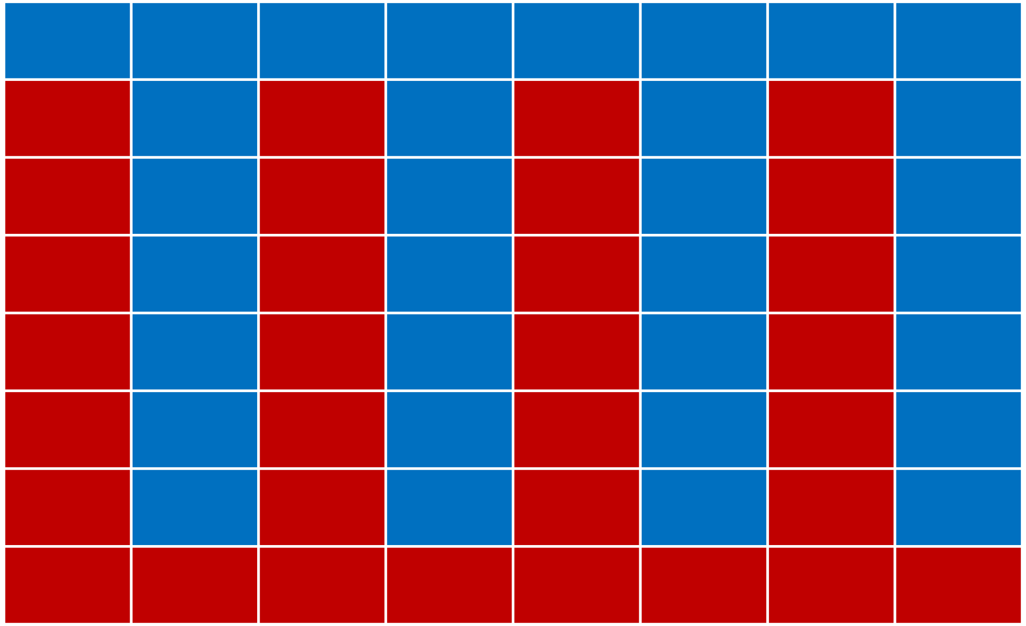
- **Solução:** a solução deste problema consiste em pintar a matriz de uma forma conveniente, em que, a princípio, não haja nenhuma intersecção, mas que elas possam ser adicionadas facilmente.
- Para isso, podemos pintar a primeira linha de azul, a última de vermelho, e nas outras linhas intercalar a cor por coluna

F – AND Grid (AtCoder – agc004_c)



F – AND Grid (AtCoder – agc004_c)

- Com esta matriz, podemos pintar uma célula roxa em qualquer posição, acrescentando a cor da região vizinha, que sempre se manterá conexa.
- A exceção são algumas células da borda, porém já sabemos que não há células roxas nestas células



F – AND Grid (AtCoder – agc004_c)