**2.DESCRIÇÂO**

Foram vendidos todos os bilhetes disponíveis para um concerto na Casa da Música. Cada bilhete tem um lugar específico. Alguns grupos de pessoas (p.ex. amigos ou familiares) não conseguiram lugares contíguos, sujeitando-se a ficarem em lugares dispersos da sala.

Pretende-se obter o menor conjunto de trocas necessárias de modo a que todos os grupos de pessoas fiquem em lugares contíguos. As mudanças a efetuar devem igualmente ter o menor impacto possível, isto é, as pessoas a mudar devem sê-lo para o lugar mais próximo possível que permita obter uma solução válida.

**3.ABORDAGEM**

Para resolver este problema em Prolog foi usada uma só lista para representar a *distribuição de público inicial,* pois para pessoas do mesmo grupo ficarem em lugares contíguos não é necessário ter em consideração as restantes filas.

Assim, se a tabela seguinte representar o público, os números o número de cada lugar e as cores os diferentes grupos, a lista tida em consideração para resolver o problema será:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |

*InputGroups([1,1,2,2,3,3,3,2,2,1,3,3])*

Ou seja, cada grupo diferente está representado com um número na lista de input no índice correspondente ao número do lugar.

**3.1 VARIÁVEIS DE DECISÂO**

A solução do problema vem na forma de 2 listas que representam a *distribuição de público final*: uma para representar a distribuição dos grupos e outra que nos diz qual o lugar inicial da pessoa atualmente naquele lugar. Assim a solução para a distribuição a cima seria:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | **10** |
| **4** | **3** | **8** |
| **9** | **5** | **6** |
| **7** | 11 | 12 |

*OutputGroups([1,1,1,2,2,2,2,3,3,3,3,3])*

*OutputIndexs([1,2,10,4,3,8,9,5,6,7,11,12])*

O tamanho das duas listas é igual ao tamanho da lista de Input. O domínio da OutputGroups é de 1 até ao número de grupos e da OutputIndexs é de 1 até tamanho da lista.

**3.2 RESTRIÇÕES**

*1. Os elementos da lista OutputIndexs têm que ser distintos.*

*2. OutputGroups[i] = InputGroups[j] /\ j = OutputIndexs[i], ou seja, na lista OutputGroups o elemento na posição i tem que ser o elemento que está na posição j na InputGroups, sendo a posição j o elemento na posição i na lista OutputIndexs*.

**3.3 FUNÇÃO DE AVALIAÇÃO**

*Lista de Distâncias*

*Lista de valores de deslocamento*

*Soma das Distâncias*

*Soma dos valores de Deslocamento*

*Número de Deslocamentos*

*Min #= TotalDifference + NumOfChanges + TotalDistance\*5*

**3.4 ESTRATÉGIA DE PESQUISA**