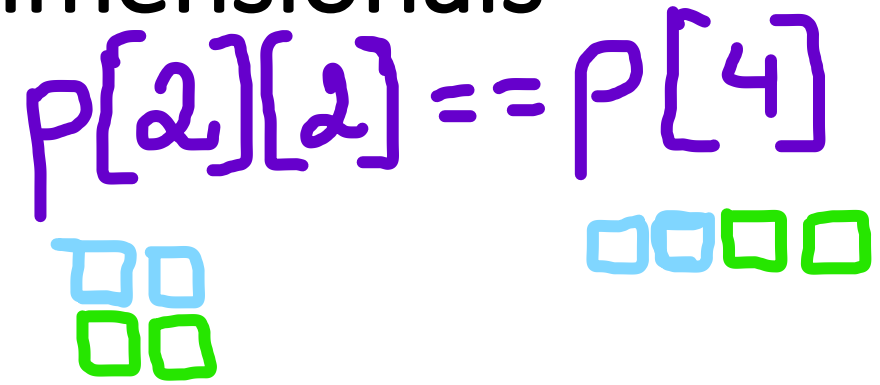
Alocação de Arrays Multidimensionais



Solução 1: usando array unidimensional

```
int i, j, linhas = 3; colunas = 2;
int *p = (int*) malloc( linhas * colunas * sizeof(int) );
for (i = 0; i < linhas; i++)
  for (j = 0; j < colunas; j++)
    p[i * colunas + j] = i+j;
for (i = 0; i < linhas; i++) {
  for (j = 0; j < colunas; j++)
    printf(" %d ", p[i * colunas + j]);
  printf("\n");
}
free(p);
```

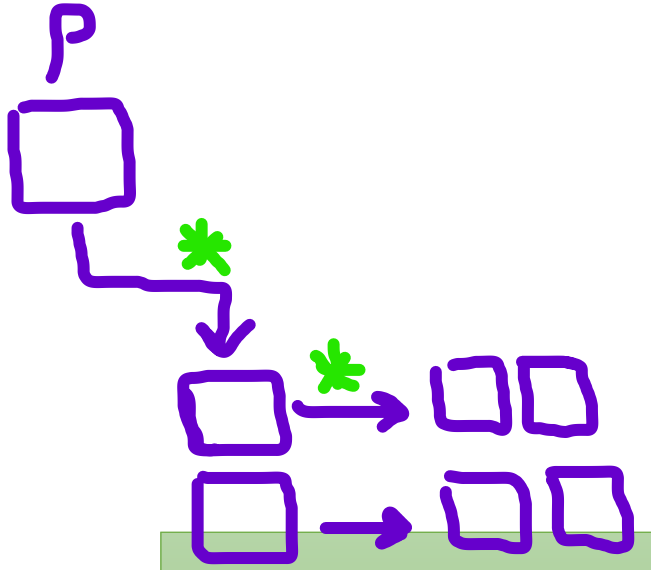
p[3][3][3] = p[27]

É possível alocar e manipular um array multidimensional da mesma forma que um array unidimensional porque um array multidimensional é armazenado na memória linearmente.

Alocação de Arrays Multidimensionais

Solução 2: usando ponteiro para ponteiro

```
int i, j, linhas = 3; colunas = 2;
int **p; //cada asterisco equivale à uma dimensão
p = (int**) malloc( linhas * sizeof(int*) );
for (i = 0; i < linhas; i++) {
  p[i] = (int*) malloc( colunas * sizeof(int) );
  for (j = 0; j < colunas; j++)
    p[i][j] = i+j;
}
for (i = 0; i < linhas; i++) {
  for (j = 0; j < colunas; j++)
    printf(" %d ", p[i][j]);
  printf("\n");
}
for (i = 0; i < colunas; i++)
  free(p[i]);
free(p);
```



Essa solução permite usar colchetes para cada dimensão.