

## 2024/10/30 資料結構深碗課程作業

41243103 林采儀

我認為拓展陣列大小的時候使用 $size*=2$ 可能會比較妥當。

因為每次增加2倍可以減少重新分配跟copy的次數，降低時間成本，

也可以確保拓展後有足夠空間可以使用

```
#include<iostream>
using namespace std;

//將陣列arr裡所有成員輸出
void print_arr(int* arr, int size)
{
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

//將原始陣列擴增為2倍大
void change_size(int*& arr, int &size)
{
    int* new_arr = new int[size * 2]; //new一塊比原始size大2倍的陣列
    int* recycler = arr; //宣告整數指標recycler，並且指向arr的位置
    copy(arr, arr + size, new_arr); //將arr內容複製到new_arr
    //將new_arr未被arr使用的陣列初始為0
    for (int i = size; i < size * 2; i++) new_arr[i] = 0;
    size *= 2; //將size陣列大小更新為原先的兩倍
    delete[] recycler; //將recycler指向的arr舊陣列回收
    arr = new_arr; //將arr的位置指向new_arr
}

int main()
{
    int size = 0; //初始陣列size=0
    cout << "請輸入原始陣列size:";
```

```
cin >> size;    //輸入陣列大小
int* arr = new int[size];    //依照輸入大小new一個陣列arr
for (int i = 0; i < size; i++)    //將arr陣列填入數值
    arr[i]=i;

cout << "size為"<<size<<"的陣列:" << endl;
print_arr(arr, size);

change_size(arr, size);
cout << "size為" << size << "的陣列:" << endl;
print_arr(arr, size);

return 0;
}
```