Find the largest sub array with equal number of 0's and 1's

```
#include <stdio.h>
#include <stdlib.h>

void findLargestSubArray(int *arr, int size)
{
        int maxSize = -1, startIndex, sumLeft[size], min = arr[0], max = arr[0], index;
        sumLeft[0] = (arr[0] == 0)? -1: 1;
   for (index = 1; index < size; index++)
   {
      sumLeft[index] = sumLeft[index - 1] + ((arr[index] == 0)?
                -1: 1);
      if (sumLeft[index] < min)
         min = sumLeft[index];
      if (sumLeft[index] > max)
         max = sumLeft[index];
   }
   int hash[max - min + 1];
   for(index = 0; index < max - min + 1; index++)
        hash[index] = -1;
   for (index = 0; index < size; index++)
   {
        if (sumLeft[index] == 0)
      {
        maxSize = index + 1;
        startIndex = 0;
      }
      if (hash[sumLeft[index] - min] == -1)
         hash[sumLeft[index] - min] = index;
      else
      {
         if ((index - hash[sumLeft[index] - min]) > maxSize)
         {
            maxSize = index - hash[sumLeft[index] - min];
            startIndex = hash[sumLeft[index] - min] + 1;
         }
      }
   }
   if (maxSize == -1)
      printf("No such subarray");
   else
      printf("Largest sub array starts from %d to %d",
         startIndex, startIndex + maxSize - 1);
}

int main()
{
        int *arr, size;
        printf("Enter size of the array\n");
        scanf("%d", &size);
```

```
	arr = (int *)malloc(sizeof(int) * size);
	printf("Enter elements in array\n");
	for(int index = 0; index < size; index++)
		scanf("%d", &arr[index]);

	findLargestSubArray(arr, size);
	return 0;
}
```

Time complexity: O(n)
Space complexity: O(n)