Giải thuật tối ưu  
int findMaxArray(int *arr*[], int *n*, int &*count\_assignment* , int &*count\_comparison* ) {

*count\_assignment* += 2;

    int count\_max = -1;

    int count = 1;

*count\_assignment* += 1;

    for (int i = 0; i < *n* - 1; i++) {

        if (*arr*[i] < *arr*[i + 1]) {

*count\_assignment* += 1;

*count\_comparison* += 1;

            count++;

            if (i == *n* - 2) {

*count\_comparison* += 1;

                if (count > count\_max) {

*count\_comparison* += 1;

                    count\_max = count;

*count\_assignment* += 1;

                }

            }

        }

        else {

*count\_comparison* += 1;

            if (count > count\_max) {

*count\_comparison* += 1;

                count\_max = count;

                count = 1;

*count\_assignment* += 2;

            }

            else {

*count\_comparison* += 1;

                count = 1;

*count\_assignment* += 1;

            }

        }

*count\_assignment* += 1;

    }

    return count\_max;

}

Giải thuật không tối ưu

int findMaxArray\_lessEfficient(int *arr*[], int *n*, int &*count\_assignment*, int &*count\_comparison*) {

*count\_assignment* += 1;

  int count\_max = 1;

*count\_assignment* += 1; //

  for (int i = 0; i < *n*; i++) {

*count\_assignment* += 1;

    int count = 1;

*count\_assignment* += 1; //

    for (int j = i + 1; j < *n*; j++) {

*count\_comparison* += 1; //

      if (*arr*[j - 1] < *arr*[j]) {

*count\_assignment* += 1; //

        count++;

      } else {

        break; //

      }

    }

*count\_comparison* += 1;

    if (count > count\_max) {

*count\_assignment* += 1;

      count\_max = count;

    }

  }

  return count\_max;

}

Kết quả so sánh A screen shot of a computer screen

Description automatically generated