Задължително домашно 1

1. Направете имплементация на алгоритъма за сортиране Bucket sort
   Алгоритъмът работи на принципа `разделяй и владей`, като:
   - елементите от масива се разпределят в отделни "кофи"(от там идва името на алгоритъма)
   - всяка кофа се сортира вътрешно(може с друг алгоритъм, може рекурсивно с bucket sort)
   - вадят се елементите от кофите поред за да се получи сортирания масив

   За кофи ще трябва да ползвате допълнителна помощна структура по избор - масив, списък, вектор и т.н.
   Изберете критерий за разпределение на кофите. В горния линк има картинки, които показват базов вариант с отделни кофи за числата от 0 до 9, от 10 до 19, от 20 до 29 и т.н.
2. Напишете функция `unsigned int get_net_size(...)`, която като аргумент получава мрежова маска и като резултат връща броя мрежови адреси, които могат да се използват в съответната мрежа
   - маската представлява група от 4 еднобайтови числа без знак (т.е. всяко е в интервала от 0 до 255). Изберете подходят тип данни и аргумент(и), с които да ги представите
   - във всяка мрежа 2 адреса (първи и последен) са запазени за самата мрежа и за broadcast, така че не трябва да влизат във върнатия резултат

   Пример: при маска 255.255.255.248 има 3 свободни бита, които дават 8 адреса, от които използваеми са 6.
3. Направете имплементация на skip list, при която да се пазят двупосочни указатели за скокове на същия принцип като prev и next в двусвързан списък(веригите от елементи за прескачане на практика са свързани списъци).
   - това ще позволи обхождане отзад-напред
   - логиката и броят елементи за прескачане в двете посоки са еднакви
   - за да бъдете по-пестеливи откъм памет комбинирайте указателя напред с указателя назад в една променлива чрез изключващо или (XOR)