Задача C - Автобусы

В городке Урюполе только один автобусный маршрут, соединяющий вокзал с главной местной достопримечательностью — продуктовым рынком, славящимся на всю округу большим ассортиментом и низкими ценами.

В Урюполь недавно пришел поезд из соседнего городка Крыжопинска, и на автобусной остановке возле вокзала образовалась очередь из N человек, желающих попасть на рынок.

В связи с этим, для развозки пассажиров к остановке собираются подать M автобусов вместимостью D каждый. Известно, что если пронумеровать людей от 1 до N в порядке очереди, то i-й из них при посадке в автобус займет L(i) единиц объема.

Однако автобус — не единственный транспорт в Урюполе: если человек устал ждать в очереди, он может выйти из очереди, сесть на такси и тут же уехать. При этом относительный порядок оставшихся в очереди людей не меняется.

Посадка в автобусы происходит следующим образом. Автобус подъезжает к остановке, открывает переднюю дверь, и в нее заходят люди в порядке очереди. Как только для очередного человека не хватает места, автобус закрывает дверь и уезжает, после чего к остановке подходит следующий автобус (если он есть).

Поскольку зарплата водителя автобуса зависит от количества перевезенных пассажиров, водители хотят знать, какое наибольшее суммарное количество людей из очереди они могут перевезти. Помогите им.

Формат ввода:

Первая строка содержит число M ().

Вторая строка — D ().

Третья строка — N ().

Четвёртая строка — L(1) L(2) … L(N). Все .

Все входные параметры — натуральные числа.

Формат вывода:

Необходимо вывести единственное число — искомое количество людей.

Решение:

Для решения данной задачи мы можем выделить следующую подзадачу: какое максимальное количество людей, которых можно посадить в автобус, используя некоторых из людей на определенном промежутке и заняв в автобусе определенное количество мест, с учетом того, что определенное количество автобусов уже занято. Если более формализовано: result[i][j][k] – максимальное количество пассажиров, где i – количество уже занятых автобусов, j – количество людей на участке очереди, из которых мы выбираем пассажиров , k – количество занятых мест в текущем автобусе.

При динамическом заполнении результирующего массива итераторы меняются следующим образом:

i - [0, num\_of\_bus-1] – так как, когда последний автобус будет заполнен – программа завершится,

j - [1, num\_of\_people],

k – [1, size\_of\_bus].

Для определения текущего result[n][m][p], у нас есть два очевидных варианта заполнения – либо мы «садим» текущего человека (1), либо нет(2).

1) В первом случае мы заполняем текущий элемент значением элемента, когда было просмотрено на одного человека меньше, и мест было занято меньше на количество, соответствующее количеству мест, занимаемых текущим человеком, увеличенным на единицу - так как мы посадили еще одного человека. Формализовано: result[n][m][p] = result[n][m-1][p-size\_of\_people[m]].

2) Во втором случае просто записываем значение, которое мы получили при «предыдущем человеке» - result[n][m][p] = result[n][m-1][p].

Так же есть третий вариант, не столь очевидный: мы «пропускаем» текущее место(3)

3) Мы записываем значение, которые мы имели при аналогичном значении автобусов/людей и количеством мест, уменьшенным на единицу. По сути, он находит свое применение лишь в одном случае: когда мы не можем посадить нового пассажира в автобус так, что бы он поместился на текущее количество мест (то есть, при условии что p<size\_of\_people[m]). В остальных же случаях нам проверять его не нужно, так как значение когда мы "садим" пассажира будет всегда больше или равно значению "пропуска места". Формализовано - result[n][m][p] = result[n][m][p-1].

В общем, у нас есть два варианта: если мы можем посадить пассажира в автобус (if (p>=size\_of\_people[m])), то сравниваем, что нам выгоднее: сделать это или не сделать. Если же мы посадить не можем, то мы сравниваем, что нам выгоднее: значение когда мы «не садим» пассажира или событие когда мы «пропускаем место».

Результат будет хранится в последней ячейке - result[n][m][p], где i=num\_of\_bus-1, j=num\_of\_people

k = size\_of\_bus.

Переход между подзадачами лишь один: когда заполнен один автобус, для определенного количества людей, то значение в result [n+1,m,0] следующего автобуса будет равным значению result[n,m,size\_of\_bus] текущего автобуса.

Оценим сложность по памяти: задействуется трехмерный массив, следовательно, сложность по памяти является линейной – F(bus\_size\*num\_of\_people\*num\_of\_bus), что при граничных значениях дает порядка 35 Мб – при лимите в 64 Мб.

Оценим сложность по времени: при проходе по динамике в каждый элемент массива мы заходим один раз, и подсчитываем его за О(1). Следовательно, сложность по времени будет линейной, и аналогичной сложности по памяти - F(bus\_size\*num\_of\_people\*num\_of\_bus). Следовательно, наше решение является полиномиальным и, даже на граничных случаях, должно укладываться в лимит по времени с неплохим запасом.