Dynamic vector

dyn\_vec

void \*arr

size\_t elem\_size = the size of an element

size\_t size = current number of elemets

size\_t capacity = size of the array

the struct will be in the c file. And not available to the client. The implementation of the struct in c wont be available to the client

typedef of the struct will be in the header ( the functions and client need to know the struct)

functions

DynVecCreate ( number of elements, size of each elements) – create the array

DynVecDestroy(pointer to the array) – destroy the array

DynVecGetItemAddress(index of the element) - return the address of an element

DynVecPushBack – add an element to the end of the array

DynVecPopBack – removes the last element, doesn't return the element, return NULL

DynVecSize – how many elements are there now

DynVecCapacity – array size, free space and elements, total size

DynVecReserve – a request to change the capacity ( enlarge or reduce the array)

when we reduce the size to ¼ of capacity, change the capacity to half (still twice the number of elements in the array)

when we realloc, and hence change the memory , the previous pointers are invalid( when we need more memory רציף) .

TO READ

adt stack

adt queue