# LINKED LISTS VS ARRAYS

# LINKED LISTS VS ARRAYS

## LINKED LISTS

LIST SIZES ARE UNLIMITED, YOU DON'T NEED TO KNOW UP FRONT HOW MANY ELEMENTS WILL BE ADDED TO THE LINKED LIST, THE SIZE OF THE LINKED LIST CAN GROW DYNAMICALLY

## ARRAYS

ARRAYS HAVE TO BE DECLARED UPFRONT WITH THE NUMBER OF ELEMENTS IT WILL HOLD, THE SIZE OF THE ARRAY CANNOT BE INCREASED DYNAMICALLY

# LINKED LISTS VS ARRAYS

## LINKED LISTS

INSERTING A NEW ELEMENT IN A LINKED LIST IS A VERY EASY OPERATION, THE LOGICAL NEXT POINTERS HAVE TO BE REASSIGNED BUT NOT MUCH ELSE NEEDS TO BE DONE

SIMILARLY DELETING AN ELEMENT IN A LIST IS VERY EASY AND EFFICIENT

## ARRAYS

ARRAYS ARE LOCATED IN CONTIGUOUS MEMORY LOCATIONS, IN ORDER TO INSERT AN ELEMENT, THE ELEMENTS TO IT'S RIGHT HAVE TO MOVE OVER TO MAKE ROOM FOR IT. IT IS A MORE HEAVYWEIGHT OPERATION

ARRAY ELEMENTS HAVE TO BE MOVED IN THE CASE OF DELETION AS WELL

# LINKED LISTS VS ARRAYS

## LINKED LISTS

RANDOM ACCESS TO AN ELEMENT AT A SPECIFIC INDEX IN THE LINKED LIST IS NOT POSSIBLE. THE ENTIRE LIST UP-TO THAT ELEMENT HAS TO BE TRAVERSED

## ARRAYS

ARRAYS PROVIDE VERY QUICK LOOKUP FOR ELEMENTS AT SPECIFIC INDICES, SINCE THEY ARE IN CONTIGUOUS MEMORY LOCATIONS, WE KNOW EXACTLY WHERE IN MEMORY THE ELEMENT IS

# LINKED LISTS VS ARRAYS

## LINKED LISTS

EACH ELEMENT REQUIRES ADDITIONAL SPACE TO STORE A POINTER TO THE NEXT ELEMENT

## ARRAYS

NO EXTRA SPACE IS REQUIRED OTHER THAN FOR THE ACTUAL ELEMENTS WHICH MAKE UP THE ARRAY

# LINKED LISTS VS ARRAYS

## LINKED LISTS

CANNOT TAKE ADVANTAGE OF SPATIAL LOCALITY (FOR CACHING) WHEN ACCESSING THE ELEMENTS

EACH ELEMENT CAN LIVE PRETTY MUCH ANYWHERE IN MEMORY AND BE POINTED TO

## ARRAYS

AS ELEMENTS ARE IN CONTIGUOUS MEMORY LOCATIONS ACCESS TO ARRAYS TAKES SIGNIFICANT ADVANTAGE OF SPATIAL LOCALITY IN CACHES

THIS CAN BE A SIGNIFICANT PERFORMANCE IMPROVEMENT

## USE LINKED LISTS WHEN:

YOU HAVE A LARGE NUMBER OF INSERT OR DELETE OPERATIONS TO PERFORM

YOU HAVE NO IDEA HOW LARGE YOUR LIST MIGHT BE

## USE ARRAYS WHEN:

READ OPERATIONS NEED TO BE EXTREMELY FAST AND YOU HAVE RELATIVELY FEW UPDATES TO THE ARRAY

YOU REQUIRE RANDOM ACCESS TO ARRAY ELEMENTS