

For Wednesday

- Read Weiss, chapter 7, section 7
- Homework:
 - Weiss, chapter 7, exercises 4, 11, 12

Exam 1

- Next Monday
- Covers material through mergesort

Programming Assignment 2

- Any questions?

Shellsort

- What's the concept?
- h-sorting
- increment sequence
- Shell's sequence $\{1, 2, 4, 8, \dots\}$
- Hibbard's sequence $\{1, 3, 7, 15, \dots\}$
- best sequence known $\{1, 5, 19, 41, 109, \dots\}$

Performance of Shellsort

Heapsort

- What's the concept?

Heapsort

- Use max heaps instead of min heaps
- Use BuildHeap to turn the array into a heap
- Use deleteMax to remove items from the beginning of the array, continually moving them to the end of the tree

Performance of Heapsort

Mergesort

- What's the concept?

Mergesort

```
void MergeSort(int arr[], int temp[], int left,
               int right)
{
    if (left < right) {
        int center = (left + right) / 2;
        // sort left half
        MergeSort(arr, temp, left, center);
        // sort right half
        MergeSort(arr, temp, center+1, right);
        // merge left and right halves
        Merge(arr, temp, left, center+1, right);
    }
}
```

```
void Merge(int arr[], int temp[], int curLeft,
           int curRight, int endRight)
{
    int endLeft = curRight - 1;
    int curTemp = curLeft;
    int numElems = endRight - curLeft + 1;
    // Main loop
    while (curLeft <= endLeft && curRight <= endRight)
        if ( arr[curLeft] <= arr[curRight])
            temp[curTemp++] = arr[curLeft++];
        else
            temp[curTemp++] = arr[curRight++];
    while (curLeft <= endLeft) // finish left
        temp[curTemp++] = arr[curLeft++];
    while (curRight <= endRight) // finish right
        temp[curTemp++] = arr[curRight++];
    // copy back to arr
    for (int i = 0; i < numElems; i++, endRight--)
        arr[endRight] = temp[endRight];
}
```

Performance of Mergesort

Bottom-up Mergesort