What questions do you have?

Decrease by a constant factor

Decrease by a variable amount

SOME MORE DECREASE-AND-CONQUER ALGORITHMS

Insertion Sort on Steroids

SHELL'S SORT – A QUICK RECAP

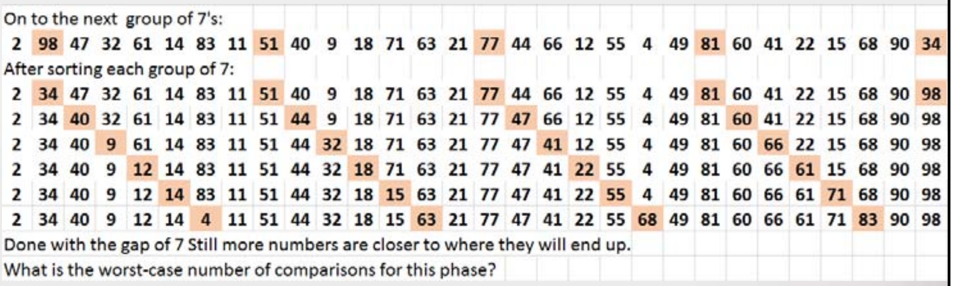
Shell's Sort

- We use the following gaps: 7, then 3, then 1 (last gap must always be 1):

21	98	47	32	61	14	83	11	51	40	9	18	71	63	90	77	44	66	12	55	4	49	81	60	41	22	15	68	2	34
Sort first 7th using insertion sort:																													
21	98	47	32	61	14	83	11	51	40	9	18	71	63	90	77	44	66	12	55	4	49	81	60	41	22	15	68	2	34
Insert 11																													
11	98	47	32	61	14	83	21	51	40	9	18	71	63	90	77	44	66	12	55	4	49	81	60	41	22	15	68	2	34
Insert 90 (nothing moves), then insert 49																													
11	98	47	32	61	14	83	21	51	40	9	18	71	63	49	77	44	66	12	55	4	90	81	60	41	22	15	68	2	34
Insert 2																													
2	98	47	32	61	14	83	11	51	40	9	18	71	63	21	77	44	66	12	55	4	49	81	60	41	22	15	68	90	34
Note that shaded numbers are now much closer to their final positions.																													

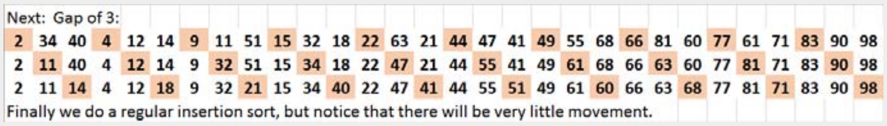
- Next, do the same thing for the next group of 7ths

Shell's sort 2



On to the next group of 7's:																													
2	98	47	32	61	14	83	11	51	40	9	18	71	63	21	77	44	66	12	55	4	49	81	60	41	22	15	68	90	34
After sorting each group of 7:																													
2	34	47	32	61	14	83	11	51	40	9	18	71	63	21	77	44	66	12	55	4	49	81	60	41	22	15	68	90	98
2	34	40	32	61	14	83	11	51	44	9	18	71	63	21	77	47	66	12	55	4	49	81	60	41	22	15	68	90	98
2	34	40	9	61	14	83	11	51	44	32	18	71	63	21	77	47	41	12	55	4	49	81	60	66	22	15	68	90	98
2	34	40	9	12	14	83	11	51	44	32	18	71	63	21	77	47	41	22	55	4	49	81	60	66	61	15	68	90	98
2	34	40	9	12	14	83	11	51	44	32	18	15	63	21	77	47	41	22	55	4	49	81	60	66	61	71	68	90	98
2	34	40	9	12	14	4	11	51	44	32	18	15	63	21	77	47	41	22	55	68	49	81	60	66	61	71	83	90	98
Done with the gap of 7 Still more numbers are closer to where they will end up.																													
What is the worst-case number of comparisons for this phase?																													

Shell's sort 3



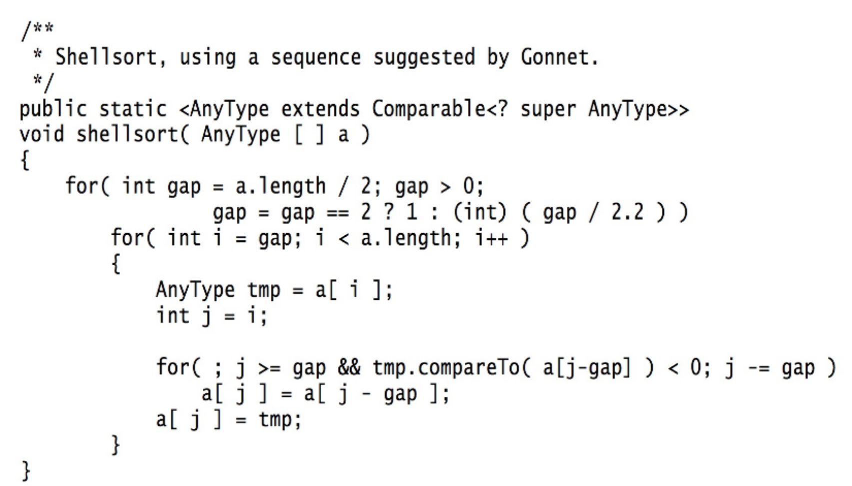
Next: Gap of 3:

2	34	40	**4**	12	14	**9**	11	51	**15**	32	18	**22**	63	21	**44**	47	41	**49**	55	68	**66**	81	60	**77**	61	71	**83**	90	98
2	**11**	40	4	**12**	14	9	**32**	51	15	**34**	18	22	**47**	21	44	**55**	41	49	**61**	68	66	**63**	60	77	**81**	71	83	**90**	98
2	11	**14**	4	12	**18**	9	32	**21**	15	34	**40**	22	47	**41**	44	55	**51**	49	61	**60**	66	63	**68**	77	81	**71**	83	90	**98**

Finally we do a regular insertion sort, but notice that there will be very little movement.

- Why bother, if we are going to do a regular insertion sort at the end anyway?
- Analysis?
- Why would this be an inferior gap sequence?
  36, 12, 3, 1
- https://www.youtube.com/watch?v=CmPA7zE8mx0

Code from Weiss book



```
/**
 * Shellsort, using a sequence suggested by Gonnet.
 */
public static <AnyType extends Comparable<? super AnyType>>
void shellsort( AnyType [ ] a )
{
    for( int gap = a.length / 2; gap > 0;
                 gap = gap == 2 ? 1 : (int) ( gap / 2.2 ) )
        for( int i = gap; i < a.length; i++ )
        {
            AnyType tmp = a[ i ];
            int j = i;

            for( ; j >= gap && tmp.compareTo( a[j-gap] ) < 0; j -= gap )
                a[ j ] = a[ j - gap ];
            a[ j ] = tmp;
        }
}
```

MORE DECREASE AND CONQUER EXAMPLES

Decrease by a constant factor

- **Examples that we have already seen:**
  - Binary Search
  - Exponentiation (ordinary and modular) by repeated squaring
  - Multiplication à la Russe (The Dasgupta book that I often used for the first part of the course calls it "European" instead of "Russian")
    - Example

11	13
5	26
~~2~~	~~52~~
1	104
	143

Then strike out any rows whose first number is even, and add up the remaining numbers in the second column.

Fake Coin Problem

- We have n coins
- All but one have the same weight
- One is lighter
- We have a balance scale with two pans.
- All it will tell us is whether the two sides have equal weight, or which side is heavier
- What is the minimum number of weighings that will guarantee that we find the fake coin?
- Decrease by factor of two?

Decrease by a variable amount

- Search in a Binary Search Tree
- Interpolation Search
  - See Levitin, pp190-191
  - Also Weiss, Section 5.6.3
  - And class slides from Session 12 (Winter, 2017)

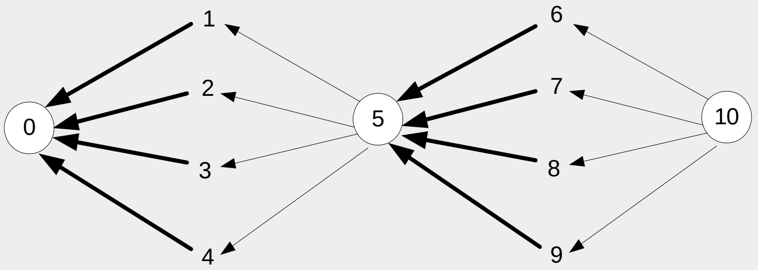
Median finding

- Find the k^{th} smallest element of an (unordered) list of n elements
- Start with quicksort's partition method
- Informal analysis

One Pile Nim

- There is a pile of n chips.
- Two players take turns by removing from the pile **at least 1** and **at most m** chips. (The number of chips taken can vary from move to move.)
- The winner is the player that takes the last chip.
- Who wins the game – the player moving first or second, if both players make the best moves possible?
- It's a good idea to analyze this and similar games "backwards", i.e., starting with $n = 0, 1, 2, \ldots$

Graph of One-Pile Nim with *m* = 4



- Vertex numbers indicate n, the number of chips in the pile.
  - The losing positions for the current player are circled.
  - Only winning moves from a winning position are shown.
- **Generalization:** The player who moves first wins iff n is not a multiple of 5 (more generally, m+1);
  - The winning move is to take n mod 5 (n mod (m+1)) chips.

Multi-Pile Nim

- There are multiple piles of chips. Two players take turns by removing from any single pile at least one and at most all of that pile's chips. (The number of chips taken can vary from move to move)
- The winner is the player who takes the last chip.
- What is the winning strategy for 2-pile Nim?
- For the general case, consider the "Nim sum", $x \oplus y$, which is the integer obtained by bitwise XOR of corresponding bits of two non-negative integers x and y.
- What is $6 \oplus 3$?

Multi-Pile Nim Strategy

- Solution by C.L. Bouton:
- The first player has a winning strategy iff the nim sum of the "pile counts" is not zero.
- **Let's prove it.** Note that ⊕ is commutative and associative.
- Also note that for any non-negative integer k, k⊕k is zero.

Multi-Pile Nim Proof

- **Notation:**
  - Let $x_1, \dots ,x_n$ be the sizes of the piles before a move, and $y_1, \dots ,y_n$ be the sizes of the piles after that move.
  - Let $s = x_1 \oplus \dots \oplus x_n$, and $t = y_1 \oplus \dots \oplus y_n$.
- **Observe:** If the chips were removed from pile k, then $x_i = y_i$ for all $i \neq k$, and $x_k > y_k$.
- **Lemma 1:** $t = s \oplus x_k \oplus y_k$.
- **Lemma 2:** If $s = 0$, then $t \neq 0$.
- **Lemma 3:** If $s \neq 0$, it is possible to make a move such that $t=0$. [after proof, do an example].
- Proof of the strategy is then a simple induction. (It's a HW problem)
- **Example:** 3 piles, containing 7, 13, and 8 chips.