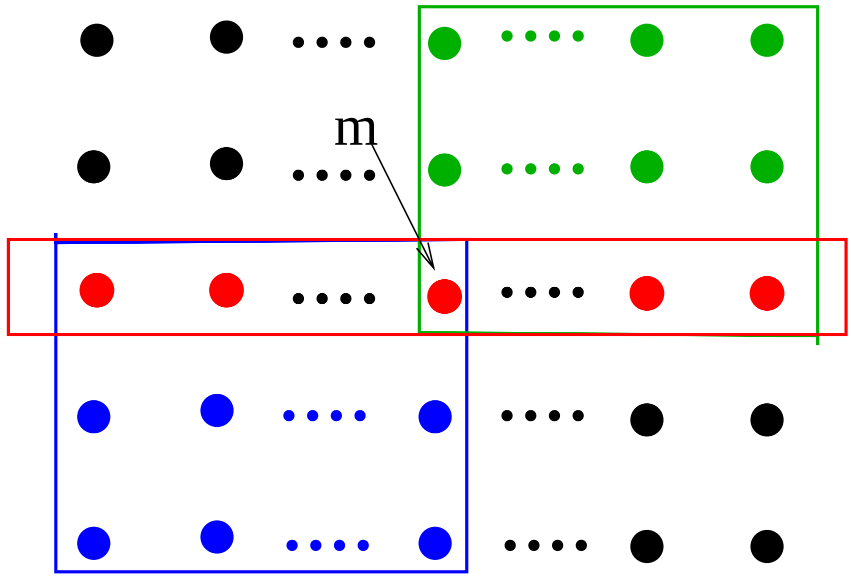
Median Finding

- Partition the items into $\lceil n/5 \rceil$ groups of 5 elements each.
- Sort each group of items.
- Arrange these sorted groups as columns such that
  - the columns appear sorted according to their median values.



Median Finding

- The elements inside the green polygon are all guaranteed to be bigger than median of medians
- The median of median values must be found to (determine the blue/green items) – requires $T(\lceil n/5 \rceil)$ time by recursive application.
- Elements inside the blue polygon are all guaranteed to be smaller than median of medians
- So, 3 out of 5 items in each group in the one half of the groups are less (greater) than m except for the group that includes m.
- For the group that includes m, 2 items are less (greater) than m
- Also if 5 does not divide n then last group have less than 5 items.

Running Time

- So discounting these 2 groups, if k is known to be greater (smaller) than m, then k can not be one of those $3 \times [n/5 \times (1/2) - 2] = (3n/10) - 6$ blue (green) items.
- In the worst case we need to find $k' = k - (3n/10) + 6$th item from among $7n/10 + 6$ items.
- The recurrence for time complexity is

$$
\begin{aligned}
T(n) &= T(7n/10 + 6) + T(\lceil n/5 \rceil) + O(n) \\
&\leq c(7n/10 + 6) + c\lceil n/5 \lceil + O(n) \\
&\leq c(7n/10) + 6c + cn/5 + c + O(n) \\
&= c7n/10 + cn/5 + 7c + O(n) \\
&= c9n/10 + 7c + O(n) \\
&\leq cn
\end{aligned}
$$

- We pick c to be large enough so that $c(n/10 - 7)$ is larger than O(n) for all $n > 80$.

Integer Multiplication

- Given two integers x and y, can partition the bit string representation into equal parts as shown

$$x = \boxed{\; a \;|\; b \;}$$

$$y = \boxed{\; c \;|\; d \;}$$

- Then
$$xy = (a2^{n/2} + b)(c^{n/2} + d)$$

- Thus the algorithm for computing xy we use 3 $n/2$ bit integer multiplication.

begin

  $u = (a + b) * (c + d)$; /* $(n/2) + 1$ bit numbers */
  $v = a * c$;
  $w = b * d$;
  $z = v * 2^n + (u - v - w) * 2^{n/2} + w$;

end

Cost of Algorithm

- Expressed as

$$T(n) = \begin{cases} k & \text{for } n = 1 \\ 3T(n/2) + kn & \text{for } n > 1 \end{cases}$$

- Applying Master theorem (case 1) solution is $\Theta(n^{\log 3})$
- More precisely, guess solution $T(n) = 3kn^{\log 3} - 2kn$, set $n = 2m$, and apply induction.

$$\begin{aligned} T(2m) &= 3T(m) + 2km \\ &= 3(3km^{\log 3} - 2km) + 2km \\ &= 3(2^{\log 3}km^{\log 3}) - 2k(2m) \\ &= 3k(2m)^{\log 3} - 2k(2m) \end{aligned}$$

- Thus $T(n) \leq 3kn^{\log 3}$ follow.

Integer Multiplication

- There is still some problem, namely, addition $a+b$ and $c+d$ may generate $(n/2)+1$ bit numbers.
- So the product u cannot be computed as directly as recursive application of the algorithm.
- To get around this problem, write

$$a+b = a_1 2^{n/2} + b_1 \text{ and } c+d = c_1 2^{n/2} + d_1$$

- Now the product $(a+b)(c+d)$ can be expressed as

$$a_1 c_1 2^n + (a_1 d_1 + b_1 c_1) 2^{n/2} + b_1 d_1$$

- a_1, c_1 are 1 bit numbers.
- But $b_1 d_1$ can be obtained by $n/2$ bit recursive integer multiplication.

Example

- Let $x = 3141$, and $y = 5927$, $a = 31$, $b = 41$, $c = 59$ and $d = 27$.
- So $a + b = 72$, $c + d = 86$.

$$\begin{aligned} u &= (a+b)(c+d) = 72 * 86 = 6192 \\ v &= ac = 31 * 59 = 1829 \\ w &= bd = 41 * 27 = 1107 \end{aligned}$$

- Therefore

$$\begin{aligned} xy &= 18290000 + (6192 - 1829 - 1107)100 + 1107 \\ &= 18616707 \end{aligned}$$

