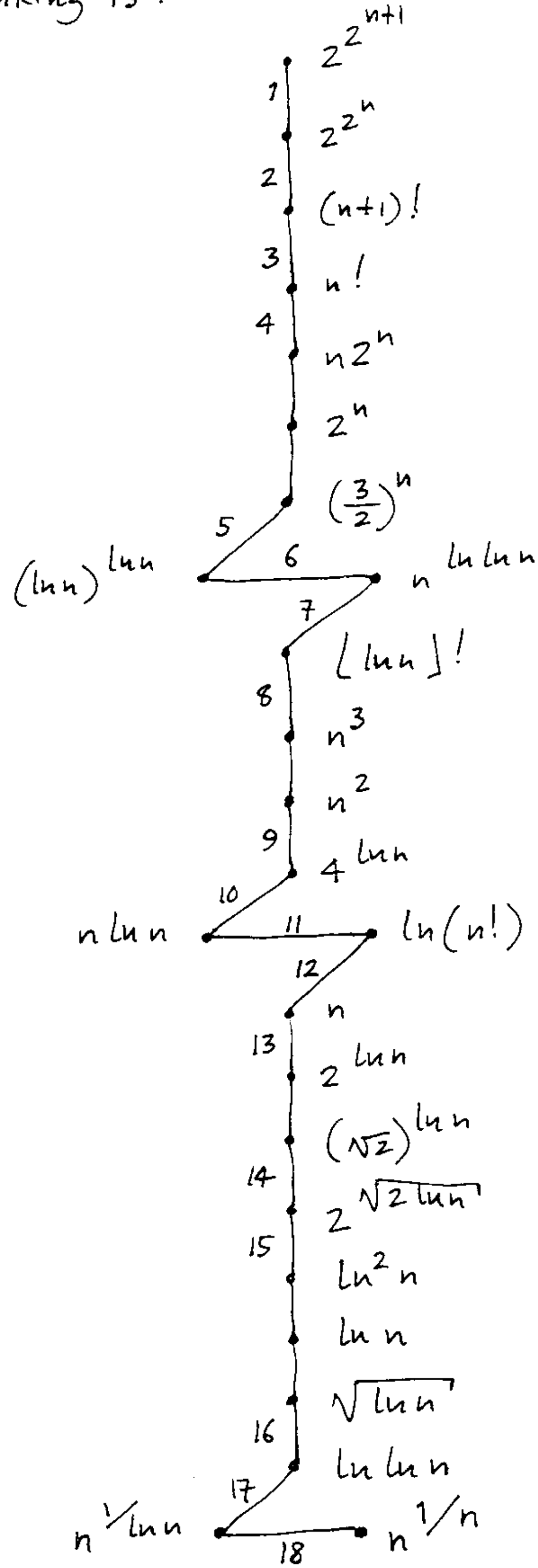
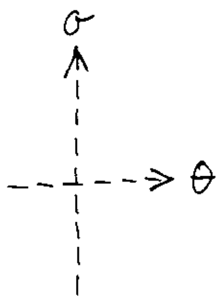
Problem (Ordering functions asymptotically)

(a) The ranking is :





Prob. cont^d

(a) cont^d

To rank the functions we use the following properties:

- Stirling's approximation

  $$n! = \Theta\left(n^{\frac{1}{2}} \left(\frac{n}{e}\right)^n\right)$$

- Polynomials vs. exponentials

  $$\forall a, b > 1 \left(n^a = o(b^n) \right)$$

- Exponential Property

  If $f(n) = 2^{h(n)}$ and $g(n) = 2^{\omega(h(n))}$ where $h(n) = \omega(1)$, then $f = o(g)$.

- Logarithm Property

  If $f(n) = \Theta(g(n))$ and both $f(n) \geq 1+\epsilon$ and $g(n) \geq 1+\epsilon$ asymptotically for some $\epsilon > 0$, then $\log f(n) = \Theta(\log g(n))$.

- Composition Property

  If $f(n) = o(g(n))$ and $h(n) = \omega(1)$, then $f \circ h = o(g \circ h)$.

Prob. cont'd

(a) cont'd

Notes:

1. $2^{2^n} = o(2^{2^{n+1}})$

- $2^{2^{n+1}} = 2^{2\cdot 2^n} = 4^{2^n}$;
- $2^n = o(4^n)$;
- Composition Prop. with $h(n) = 2^n$.

2. $(n+1)! = o(2^{2^n})$

- $(n+1)! = 2^{\lg((n+1)!)} = 2^{\lg \theta\left((n+1)^{\frac{1}{2}} \left(\frac{n+1}{e}\right)^{n+1}\right)}$ by Stirling's approx.
  $= 2^{\theta(n \log n)}$ by Logarithm Prop.;
- Exponential Prop. with $\theta(n \log n) = o(2^n)$.

3. $n! = o((n+1)!)$

- $n! = \theta(n^{\frac{1}{2}} e^{-n} n^n)$,
  $(n+1)! = \theta(n^{\frac{1}{2}} e^{-n} n^{n+1})$ by Stirling's approx.

4. $n2^n = o(n!)$

- $n2^n = 2^{\lg n} 2^n = 2^{n+\lg n} = 2^{\theta(n)}$;
- $n! = 2^{\theta(n \log n)}$ by Note 2;
- Exponential Prop. with $\theta(n) = o(n \log n)$.

5. $(\ln n)^{\ln n} = o\left(\left(\frac{3}{2}\right)^n\right)$

- $(\ln n)^{\ln n} = (2^{\lg \ln n})^{\ln n} = 2^{\theta(\log n \log\log n)}$;
- $\left(\frac{3}{2}\right)^n = (2^{\lg \frac{3}{2}})^n = 2^{\theta(n)}$;
- Exponential Prop. with $\theta(\log n \log\log n) = o(n)$.

Prob. cont^d

(a) cont^d

6. $(\ln n)^{\ln n} = \theta\left(n^{\ln \ln n}\right)$

- $(\ln n)^{\ln n} = \left(e^{\ln \ln n}\right)^{\ln n} = \left(e^{\ln n}\right)^{\ln \ln n} = n^{\ln \ln n}$.

7. $\lfloor \ln n \rfloor ! = o\left(n^{\ln \ln n}\right)$

- $\lfloor n \rfloor ! = \theta\left(\frac{n^{\frac{1}{2}}}{e^n} n^n\right)$ by Stirling's approx.

  $= o(n^n)$ by Poly^s vs. Exp^s;

- $\lfloor \ln n \rfloor ! = o\left((\ln n)^{\ln n}\right)$ by above and Composition Prop. with $h(n) = \ln n$

  $= o\left(n^{\ln \ln n}\right)$ by Note 6.

8. $n^3 = o\left(\lfloor \ln n \rfloor !\right)$

- $(e^n)^3 = 2^{\theta(n)}$;
- $n! = 2^{\theta(n \log n)}$ as in Note 2;
- $(e^n)^3 = o\left(\lfloor n \rfloor !\right)$ by above and Exponential Prop.;
- $n^3 = o\left(\lfloor \ln n \rfloor !\right)$ by above and Composition Prop. with $h(n) = \ln n$.

9. $4^{\ln n} = o(n^2)$

- $4^{\ln n} = \left(e^{\ln 4}\right)^{\ln n} = n^{\ln 4} = n^{2-\epsilon}$ where $\epsilon > 0$.

Prob. cont'd

(a) cont'd

10. $n \ln n = o(4^{\ln n})$

- $4^{\ln n} = n^{\ln 4} = n^{1+\epsilon}$ where $\epsilon > 0$;
- $\ln n = o(n^{\epsilon})$ for any $\epsilon > 0$ by Poly's vs. Exp's with $a = 1$, $b = e^{\epsilon}$, and Composition Prop. with $h(n) = \ln n$.

11. $n \ln n = \theta(\ln(n!))$

- $\ln(n!) = \theta\left(\log\left(n^{\frac{1}{2}}\left(\frac{n}{e}\right)^n\right)\right)$ by Stirling's approx. and Logarithm Prop.
  $= \theta(\theta(n \log n))$.

12. $n = o(\ln(n!))$

By Note 11.

13. $2^{\ln n} = o(n)$

- $2^{\ln n} = n^{\ln 2} = n^{1-\epsilon}$ where $\epsilon > 0$.

14. $2^{\sqrt{2 \ln n}} = o\left((\sqrt{2})^{\ln n}\right)$

- $2^{\sqrt{2 \ln n}} = 2^{\theta(\log^{\frac{1}{2}} n)}$;
- $(\sqrt{2})^{\ln n} = 2^{\theta(\log n)}$;
- Exponential Prop. with $\theta(\log^{\frac{1}{2}} n) = o(\log n)$.

Prob. cont^d

(a) cont^d

15. $\ln^2 n = o\left(2^{\sqrt{2\ln n}}\right)$

- $\ln^2 n = 2^{\Theta(\ln \ln n)}$;
- $2^{\sqrt{2\ln n}} = 2^{\Theta(\sqrt{\ln n})}$;
- $\ln \ln n = o(\sqrt{\ln n})$,

  by Poly^s vs. Exp^s with $a=1$, $b=\sqrt{e}$

  and Composition Prop. with $h(n) = \ln \ln n$.

16. $\ln \ln n = o(\sqrt{\ln n})$

- By Note 15.

17. $n^{1/\ln n} = o(\ln \ln n)$

- $n^{1/\ln n} = \left(e^{\ln n}\right)^{1/\ln n} = e = \Theta(1)$;
- $\Theta(1) = o(\ln \ln n)$ since $\lim_{n\to\infty} \ln \ln n = \infty$.

Prob. cont'd

(a) cont'd

14. $n^{1/\ln n} = \Theta\left(n^{1/n}\right)$

- By Note 17, $n^{1/\ln n} = \Theta(1)$.
- We show $n^{1/n} = \Theta(1)$ as follows:

(i) $\forall n \geq 1 \left(n^{1/n} \geq 1^{1/n} = 1 \right) \Rightarrow n^{1/n} = \Omega(1)$;

(ii) $\max\limits_{n>0} \left\{ n^{1/n} \right\} = e^{1/e} \Rightarrow n^{1/n} = O(1)$:

$$\frac{d}{dn}\left(n^{1/n}\right) = \frac{d}{dn}\left(\left(e^{\ln n}\right)^{1/n}\right) = n^{1/n}(1-\ln n)/n^2,$$

which is identically zero when $n = e$;

moreover $\left.\frac{d^2}{dn^2}\left(n^{1/n}\right)\right|_{n=e} = \left.\left(-n^{\frac{1}{n}-3}\right)\right|_{n=e} < 0$,

so the function attains its maximum at e.

(b) For any $\epsilon > 0$, consider the function

$$f(n) := \underbrace{\left(\tfrac{1}{2}(1+\sin n)\right)}_{\text{oscillates between 0 and 1}} n^{-\epsilon} + \left(1 - \tfrac{1}{2}(1+\sin n)\right) n^{\epsilon} 2^{2^{n+1}}.$$

This function touches both $n^{-\epsilon} = o(1)$ and $n^{\epsilon} 2^{2^{n+1}} = \omega\left(2^{2^{n+1}}\right)$ infinitely often, so

$$f(n) \neq \Omega(1),$$

and

$$f(n) \neq O\left(2^{2^{n+1}}\right).$$

Conjecture

Let $f(n)$ be asymptotically positive such that for all c,

$$f(n) = \omega(n^c).$$

Then for some b,

$$f(n) = \Omega(b^n).$$

Disproof

Consider the counterexample

$$f(n) = n^{\lg n}.$$

Note that

$$n^c = (2^{\lg n})^c = 2^{\Theta(\log n)},$$

$$n^{\lg n} = (2^{\lg n})^{\lg n} = 2^{\Theta(\log^2 n)},$$

$$b^n = (2^{\lg b})^n = 2^{\Theta(n)}.$$

Moreover,

$$\Theta(\log n) = o(\log^2 n),$$ (by $n = o(n^2)$ and the Composition Property using $\log n$)

and

$$\Theta(\log^2 n) = o(n).$$ (by $n^2 = o(2^n)$ and the Composition Property using $\log n$)

Thus by the Exponential Property,

$$\forall c \left(f(n) = \omega(n^c) \right),$$

but also

$$\forall b \left(f(n) = o(b^n) \right),$$

disproving the conjecture. $\square$

<u>Theorem</u> The divide-and-conquer recurrence

$$T(n) = a\,T\left(\frac{n}{b}\right) + \Theta\left(n^c \log^d n\right)$$

for constants $a > 0$, $b > 1$, $c \geq 0$, $d \geq 0$, has the solution

$$T(n) = \begin{cases} \Theta(n^{\log_b a}), & c < \log_b a; \\ \Theta(n^c \log^{d+1} n), & c = \log_b a; \\ \Theta(n^c \log^d n), & c > \log_b a. \end{cases}$$

<u>Problems</u>

- $T(n) = 4\,T\left(\frac{n}{2}\right) + n^2\sqrt{n}$ $\quad \begin{cases} a = 4 \\ b = 2 \\ c = 5/2 \\ d = 0 \end{cases}$

  $c > \log_b a$:

  $$\boxed{T(n) = \Theta(n^{5/2})}$$

- $T(n) = 3\,T\left(\frac{n}{2}\right) + n \lg n$ $\quad \begin{cases} a = 3 \\ b = 2 \\ c = 1 \\ d = 1 \end{cases}$

  $c < \log_b a$:

  $$\boxed{T(n) = \Theta(n^{\log_2 3})}$$

- $T(n) = 2\,T\left(\frac{n}{2}\right) + n \lg n$ $\quad \begin{cases} a = 2 \\ b = 2 \\ c = 1 \\ d = 1 \end{cases}$

  $c = \log_b a$:

  $$\boxed{T(n) = \Theta(n \log^2 n)}$$

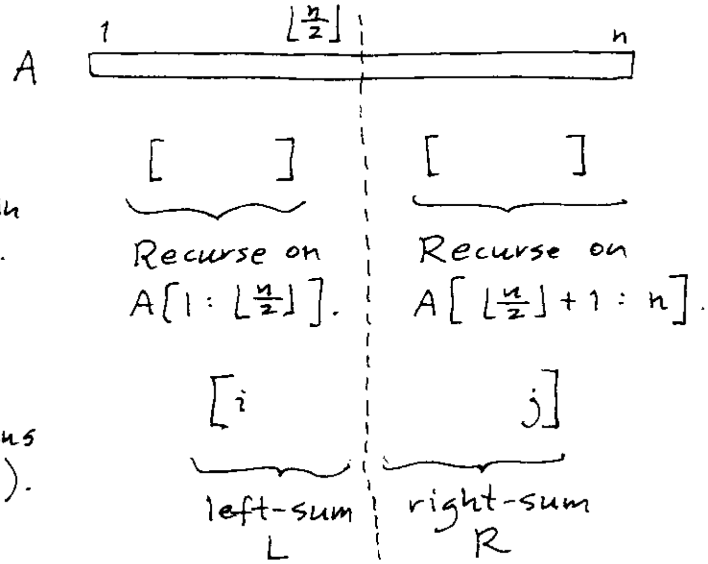
Problem (Minimum positive-sum subarray)

Given array $A[1:n]$ of real numbers, find a subarray $A[i:j]$ s.t. $\sum_{i \le k \le j} A[k]$ is

- strictly greater than zero, and
- minimum.

(a) **Proposition** Using divide-and-conquer, we can find a minimum positive-sum subarray in $\Theta(n \log^2 n)$ time.

Algorithm We split the array in half and consider where the optimal subarray might fall w.r.t. the split:



$2T(\frac{n}{2})$ — · Cases 1,2 (sol^{n} falls in one half).

· Case 3 (sol^{n} spans the split).

For Case 3, compute all $\lceil \frac{n}{2} \rceil$ possible right-sums $\sum_{\lfloor \frac{n}{2} \rfloor < k \le j} A[k]$.

$+\ \Theta(n \log n)$ — Sort these sums.

$+\ \Theta(n \log n)$ — Then for each left-sum L, find the minimum right-sum R s.t. $R > -L$ using binary search on the sorted sums.

Record the best (L, R) pair.

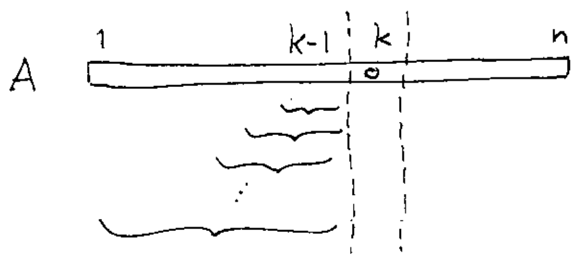
Analysis This takes time $T(n) = 2T(\frac{n}{2}) + \Theta(n \log n)$

$= \Theta(n \log^2 n)$. $\square$

Prob. cont'd (Min. pos.-sum subarray)

(b) Proposition Using an incremental strategy, we can solve the problem in $O(n \log n)$ time.

Algorithm For $k = 1, 2, \cdots, n$ we find the best solution whose right end is at k, given that this problem has been solved for $k-1$:



We maintain a balanced search tree T of intervals whose right end is at $k-1$, and a real number δ. Each elt of T is a key-item pair (x, i) where $x + \delta = \sum_{i \le j < k} A[j]$. (Initially T is empty and $\delta = 0$.)

Instead of incrementing keys in T when k is increased, we just increment δ.

```
function MinPosSumSubarray(A, n) begin
    m := ∞
    δ := 0
    T := Tree()
    for k := 1 to n do begin
        Insert (-δ, k) into T.                          • Inserts the empty interval.
        Find the elt (x, i) of T with smallest key x
          s.t. x > -(δ + A[k]).                         • Notice x + δ + A[k] > 0.
        if elt (x, i) exists then
            m := min{m, x + δ + A[k]}
        δ +:= A[k]                                      • Appends A[k] to all intervals in T.
    end
    if m < ∞ then return m else return ⊥
end
```

O(n log n) time — O(log n) time

$\square$

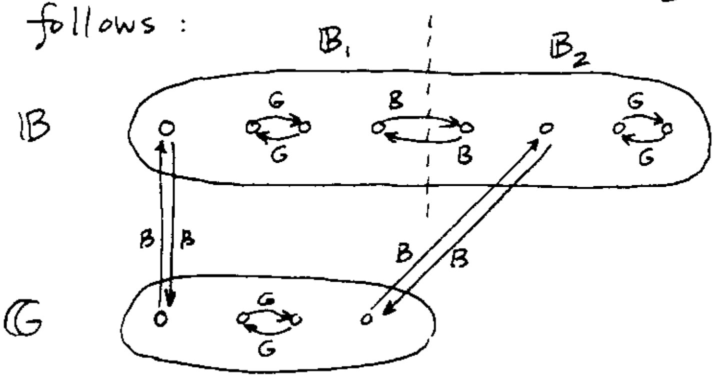
Problem (Identifying good chips by pairwise tests)

(a) Theorem For every $n > 1$ and $0 < g \le \frac{n}{2}$, there is a set $\mathbb{R}$ of pairwise test results on n chips with the following two properties:

(i) Two distinct states $\mathbb{S}$ and $\mathbb{S}'$ are consistent with $\mathbb{R}$.

(ii) States $\mathbb{S}$ and $\mathbb{S}'$ both have g good chips.

Proof Given n and g, we construct $\mathbb{R}$ and state sets $\mathbb{S}$ and $\mathbb{S}'$ as follows. Let $\mathbb{S} = (\mathbb{G}, \mathbb{B})$, where $\mathbb{G}$ and $\mathbb{B}$ are the sets of good and bad chips, with $|\mathbb{G}| = g$ and $|\mathbb{B}| = n-g$. Split $\mathbb{B}$ into $\mathbb{B}_1, \mathbb{B}_2$ where $|\mathbb{B}_1| = g$ and $|\mathbb{B}_2| = n-2g$. (Note this can be done since $|\mathbb{B}| \ge \frac{n}{2} \ge |\mathbb{G}|$.) Results $\mathbb{R}$ are as follows:



Now consider the new state $\mathbb{S}'$ obtained by exchanging each chip in $\mathbb{G}$ with exactly one chip in $\mathbb{B}_1$. Both $\mathbb{S}$ and $\mathbb{S}'$ are consistent with $\mathbb{R}$ and have exactly g good chips. $\square$

Corollary Suppose there are g good chips where $0 < g \le \frac{n}{2}$. Then no algorithm can identify the good chips in the worst case

Proof Let the test results given to the algorithm come from $\mathbb{R}$ above. Whatever set of chips the algorithm declares to be good an adversary can reveal a true state where the set is actually bad. $\square$

Problem contd.

(b) Lemma The problem of finding one good chip from among $n > 1$ chips, where more than half the chips are good, can be reduced to a nonempty problem on at most $\frac{n}{2}$ chips where more than half the chips are good, using $\Theta(n)$ pairwise tests.

Proof Arbitrarily pair off every chip with one other (except for possibly one odd chip) and test the pairs. Group the pairs by result:

$$\underbrace{G \rightleftarrows G \ \cdots \ G \rightleftarrows G}_{\lfloor \frac{n}{2} \rfloor - x \text{ pairs}} \quad \underbrace{\underbrace{B \rightleftarrows B \ \cdots \ B \rightleftarrows B} \ \underbrace{G \rightleftarrows B \ \cdots \ G \rightleftarrows B}}_{x \text{ pairs}} \quad \underset{\text{(odd chip)}}{\circ}$$

Every $\overset{B}{\underset{B}{\rightleftarrows}}$ or $\overset{G}{\underset{B}{\rightleftarrows}}$ pair contains ≥ 1 bad chip. Thus the number x of pairs of this type satisfies $x < \lceil \frac{n}{2} \rceil$ (since $> \lfloor \frac{n}{2} \rfloor$ chips are good).

Furthermore the number of good chips among $\overset{G}{\underset{G}{\rightleftarrows}}$ pairs plus the odd chip is $> \lfloor \frac{n}{2} \rfloor - x$ (since there are $> \lfloor \frac{n}{2} \rfloor$ good chips, and ≤ 1 good chip in every $\overset{B}{\underset{B}{\rightleftarrows}}$ or $\overset{G}{\underset{B}{\rightleftarrows}}$ pair).

Case n even

Arbitrarily select one chip from every $\overset{G}{\underset{G}{\rightleftarrows}}$ pair. This chooses $\frac{n}{2} - x$ chips, which is a nonempty set (as $x < \frac{n}{2}$).

This selection must contain $> \frac{1}{2}(\frac{n}{2} - x)$ good chips (since each $\overset{G}{\underset{G}{\rightleftarrows}}$ pair is either both good or both bad, so the number of good chips among $\overset{G}{\underset{G}{\rightleftarrows}}$ pairs is exactly twice the number of good chips in the selection).

This yields a nonempty set of $\leq \lceil \frac{n}{2} \rceil$ chips of which $> \frac{1}{2}$ are good.

Problem contd

(b) Proof contd

Case n odd

Subcase $\lfloor \frac{n}{2} \rfloor - x$ is even (possibly zero)

Arbitrarily select one chip from each $\overset{G}{\underset{G}{\rightleftarrows}}$ pair, plus the odd chip. This chooses $\lfloor \frac{n}{2} \rfloor - x + 1$ chips.

Suppose this selection has $\leq \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x + 1\right)$ good chips, or equivalently $\leq \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x\right)$ good chips (since $\lfloor \frac{n}{2} \rfloor - x + 1$ is odd). Then the $\overset{G}{\underset{G}{\rightleftarrows}}$ pairs plus the odd chip contain $\leq \lfloor \frac{n}{2} \rfloor - x$ good chips (i.e. at most twice the number of good selected chips), which contradicts that they must contain $> \lfloor \frac{n}{2} \rfloor - x$ good chips.

So the selection must have $> \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x + 1\right)$ good chips.

Subcase $\lfloor \frac{n}{2} \rfloor - x$ is odd

Arbitrarily select one chip from each $\overset{G}{\underset{G}{\rightleftarrows}}$ pair (without the odd chip). This chooses $\lfloor \frac{n}{2} \rfloor - x$ chips.

Suppose this selection has $\leq \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x\right)$ good chips, or equivalently $\leq \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x - 1\right)$ good chips (since $\lfloor \frac{n}{2} \rfloor - x$ is odd). Then the $\overset{G}{\underset{G}{\rightleftarrows}}$ pairs plus the odd chip contain $\leq \left(\lfloor \frac{n}{2} \rfloor - x - 1\right) + 1$ good chips, which contradicts that they must contain $> \lfloor \frac{n}{2} \rfloor - x$ good chips.

So the selection must have $> \frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor - x\right)$ good chips.

In either subcase this yields a nonempty set of $\leq \lceil \frac{n}{2} \rceil$ chips of which $> \frac{1}{2}$ are good.

$\square$

Problem contd

(c) Theorem All good chips from among n chips can be identified using $\Theta(n)$ pairwise tests, assuming $> \frac{n}{2}$ chip are good.

Proof Using the lemma from part (b), we can find one good chip by repeatedly recursing on a subproblem of at most half the size until only one chip remains. This uses

$$T(n) \leq T\left(\frac{n}{2}\right) + \Theta(n)$$
$$= O(n)$$

pairwise tests.

After the single good chip is found, we can use it to test all the other chips. This identifies all good chips using an additional $\Theta(n)$ tests, for a total of $\Theta(n)$ pairwise tests.

$\square$