Optimization

Find min or max value (or local min/max)
in state space landscape
→ objective function if max, else cost function

Hill climb (steepest ascent)
  → state = initial (select random)
  → generate nbrs
  → find best neighbour, replace curr state

Issue: gets stuck at local min.

Beam search: many different locals at once

Sim. annealing: use temperature and try hill-climb
up to max times.
  → ΔE = how much better is nbr state than curr
  → $\Delta E > 0$, replace
  → else, replace with probability $e^{\Delta E / t}$
  → on each run temp. decreases: less likely to replace
  → helps to explore nbrs that are worse → eventually get to better

$$3^0 \cdot 5^0 = 1$$

$$3^x 5^y \qquad \{(x,y) \mid x \in 2, y \in 2, \sim(x=9 \wedge y=0)\}$$

$[19, \sqrt{6}]$

$20, 81, 24, 25, 27, 30, 32, 33, 36, 39, 40, 43, 45$

L

$$38$$

$$3^2 \cdot 7 \qquad LB \; x/y \qquad 8^2 \cdot 5^1$$

$$UB \; x/y$$

$$\hookrightarrow LB \; x \qquad UB \; y$$

$$LB \; y \qquad UB \; y$$

$$6$$

$$\text{floor}$$

$$\frac{x}{15 x} \cdot \cancel{15} \; x \qquad \hookrightarrow \text{cell}(\; \frac{15(x)}{15(3)}\;)$$

$$x \, l_{sy} x + \frac{x^2}{15 x} \quad \rightarrow \text{slove of gen} \qquad \overline{} \; 5$$

$$+9$$

$$1_{25 \cdot 15} \qquad 1\, 3\quad\underbrace{\;5\;6\;9\;12\;15\;18\;21\;27\;}_{}\;$$

$$1_{5_3}(x 27 \cdot 1_{5_3}(x)) \qquad \cdots \overline{27}$$

$$1\;3\;9\;27$$

$$3 \times 5 \qquad 5 \quad 25 \quad 27$$

$$/15 \qquad 45$$

$1, 20$

$\hookrightarrow 3, 5, 6, 9, 12, 18, 15, 18, 20 \qquad 3^2$
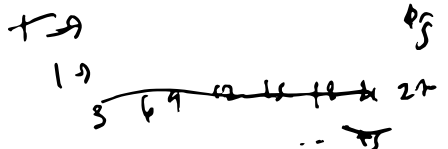
$$x = 9 \qquad\qquad\qquad 3^x 5^y \; \text{ih } 1 \cdots 20$$

$$3^0 = 1 \qquad 3^0 \cdot 5^0 = 1 \qquad 3^1 \cdot 5^0 = 3$$

$$3^0 \cdot 5^1 = 5 \qquad 3^1 \cdot 5^1 = 15$$

$$3^1 \cdot 5^2 = 25 \qquad 3^1 \cdot 5^2 = \cdots$$

$$\text{ceil}(\; l_{5_5}(\frac{end}{3^0})\;)) = 12 \quad (1, 57) \qquad 3^2 \cdot 5^0 = 9$$

$$1, 9$$

$$\text{ceil}(l_{5_5}(\frac{20}{3^1})) = 2 \qquad (3, 15)$$

$$\frac{9}{3^2}$$

$$\text{ceil}(l_{5_5}(\frac{20}{3^2})) = 1 \qquad (9)$$

$$[1, 9] \qquad 3, 5, 9$$

$$3^0 \cdot 5^0 = 1 \qquad 3^1 \cdot 5^0 = 3 \qquad 3^2 \cdot 5^0 = 9$$

$$5^0 \cdot 5^1 = 5$$

$$\text{ceil}(l_{5_5}(\frac{end\,9}{3_0})) = 2 \quad e(1,5)$$

$$\text{ceil}(l_{5_5}(\frac{9}{3^1})) = 1 \quad \cancel{(9)} \; (3)$$

$$\text{ceil}(l_{5_5}(\frac{9}{3^2})) = 1 \; (9)$$

```
find-test                    related cmds:
  └ a.txt                      → tree, find, grep
  └ find2
    └ a.txt

find . -name "a.txt"
  => ./find2/a.txt
     ./a.txt
```

---

1. Take url

2. get HTML → Js dom

3. Use DFS or BFS
   → depth-limited search
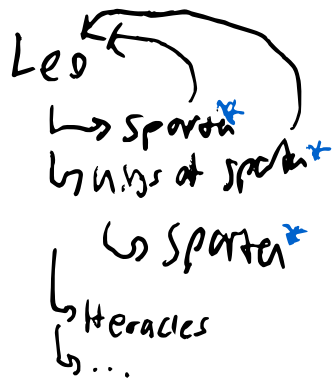   → add to vis set with depth

1. Infinite dir ✓
2. mult. goals ✓
3. depth 1: Leo/Sparta
            Leo/kings of Sparta
4. first match: Leo/Sparta ✓
5. limit 2: Leo/Sp,
            Leo/kings of

```
Leo
 └→ Sparta ✗
 └ kings of Sparta ✗
    └→ Sparta ✗
 └ Heracles
 └ ...
```

1. Depth limit
   or
2. matches limit
3. First match
   → unspecified order

find w curl > Sparta - title

P async procs exec. m tasks

How many tasks? m is quite big

(p, s, depth)

(p,s,e) = (p2, d-1)
          ...   ...
with task/work-stealing?

Async task alloc: "write-all"

→ P processes, m tasks
→ P procs vary in speed etc. (heterog)
→ need decentralized scheduling → comm thru shared mem
→ ends: P procs produce cert. that tasks done
procs select random task from task set to exec.
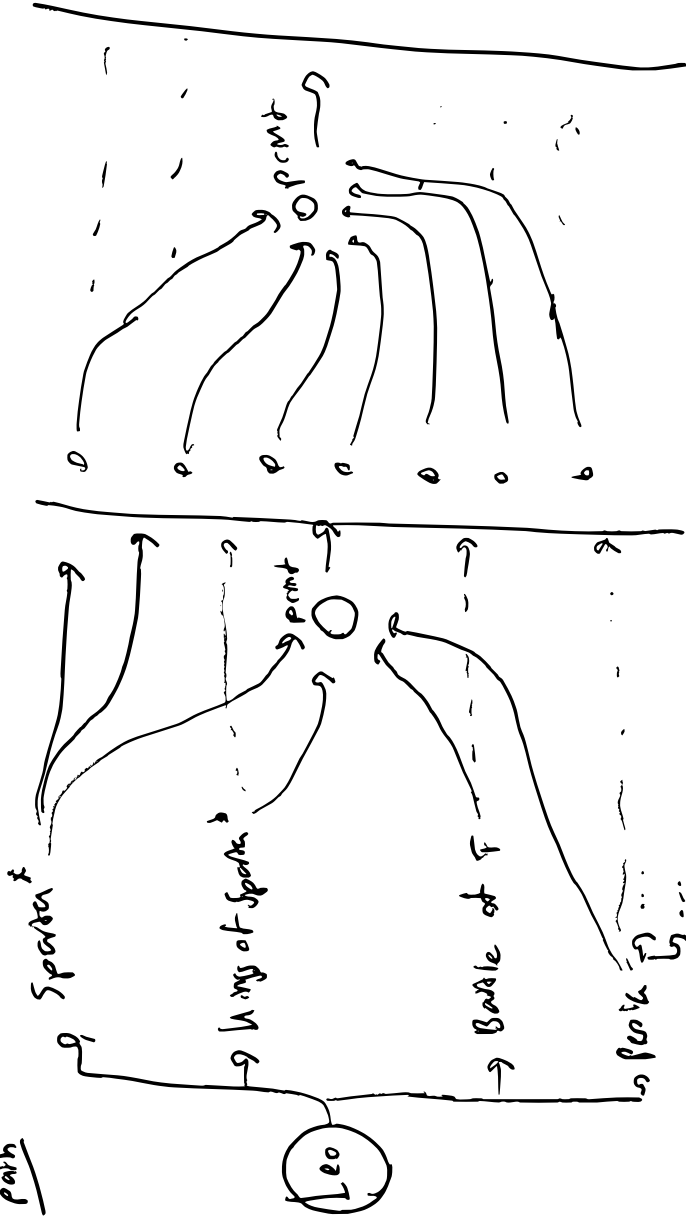
Rust   1. Identify dependence → draw task graph
             → where is sync needed?
        2. Write pseudocode for BFS based on
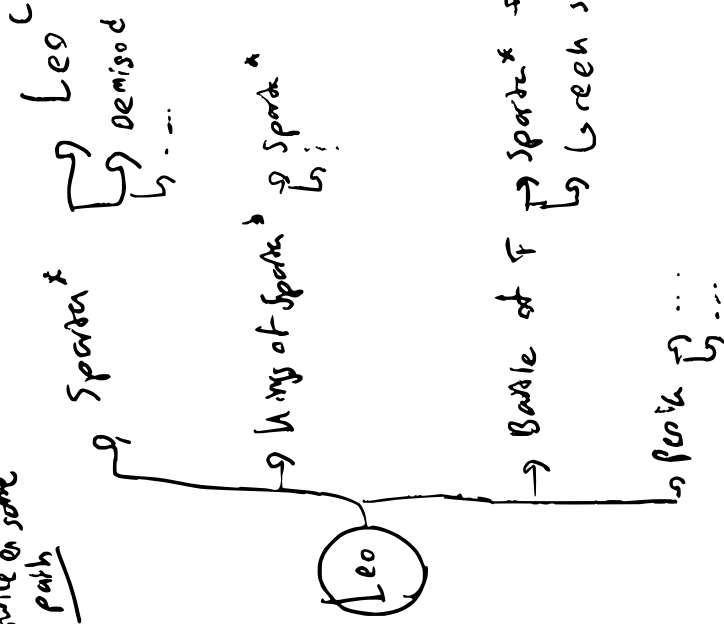   Nars - identify changes
        3. Benchmark
        → critical path (span)   avg concurrency
           work   work/span
```

KP:

C: when node
built on some
path

→ Spartan*

⌐ Leo^c
└ Demigod
   └ ...

→ Uprising of Spartan* → Spartan*
                           └ ...

Leo

→ Battle of T ⌐→ Spartan* → Battle of T ^c
               └→ Green states → Spartan*

→ Death ⌐
        └ ...

w gch: $O(b^d)$
→ b = max branch factor
→ d = critical path | en

Spam, $O(d)$

Request graph (data)
→ assigned to remotely

C: when node
built on same
path

Speaker *

things of Speak *

Battle of T

Leo

...feat

Pcmt

Pcmt

Pcmt Graph (lax 2)

wgch: $O(b^2d)$
→ b = max branch factor
→ d = critical path len

span, $O(d)$

$([[407]], 47, [[so]]])$ $(x7)$