Kerge:

1. Käsureapõhine trips-traps-trull
2. Typeracer (sarnaselt <https://play.typeracer.com/> )
3. Diff (näiteks kuidas 2 tekstifaili teineteisest erinevad, mis real, mille poolest)
4. Poomismäng
5. Mullisort (Algoritmi realisatioon kirjeldusega)

Keskmine:

1. Dijkstra algoritm (Algoritmi realisatsioon kirjeldusega)
2. Lineaarvõrrandisüsteemide lahendaja
3. Mõne geneetilise kauguse arvutamine. Raskendav boonus: meetodi kiirendamine Intel X86\_64 vektorinstruktsioonidega.
4. Mõni kasulik algoritm mis on teostatud C++ ja saab kutsuda mõnest teisest keelest (näiteks R, Python, Javascript/Node.js, etc). Sellist lähenemist kasutatakse sageli, et aeglane osa dünaamilises keeles kirjutatud skriptist palju kiiremaks teha.

Raske:

1. Teise protsessi mälu ruumis tegutsemine ( mälu lugemine/ muutmine ). Oleneb operatsioonisüsteemist, windowsis (Abiks “Windows.h” päis).
2. Hashcracker (Etteantud parooliräsi brute-forcemine), võib toetada ka ainult ühte algoritmi
3. Boids -- simuleerimine ja visualiseerimine
4. Väga lihtne software only ujukoma (liitmine, korrutamine, pöördarv, ruutjuur)
5. Implicit Runge-Kutta diferentisaalvõrrandite lahendaja (üldine numbriline lahendaja mis võtab sisendiks suvalise funktsiooniobjekti). Tungivalt soovitatav varasem kogemus numbriliste meetoditega. Võib kasutada välist lineaarvõrrandisüsteemide lahendajat.
6. Valguse simuleerimine ja visualiseerimine 2D ruumis (vaata <https://benedikt-bitterli.me/tantalum/tantalum.html>). Äärmiselt mahukas teema kui ei ole juba füüsika ja selle simuleerimisega tuttav.
7. Minesweeperi lahendaja. Peaks suutma lahendada juhuslikult genereeritud intermediate tasemega puslesid keskmiselt enam kui veerand juhtudel.
8. Jaapani mõistatuste lahendaja. Suudaks mõistlikus ajas lahendada inimestele mõeldud ülesandeid kuni suuruses 50 korda 50. Puslesid peaks saama anda sisendina lihtsas tekstilises formaadis (ei pea vastama täpselt lingis toodule).