1 **/\***

2 **\* Ovo je prijepis mog hibridnog algoritma razvrstavanja da se može**

3 **\* pokrenuti na JavaScriptinoj virtualnoj mašini. Kako bi se AEC-om mogla**

4 **\* ciljati JavaScriptina virtualna mašina, napravio sam novi compiler.**

5 **\* Ovaj puta je pisan u C++-u, radio sam i novi parser i novi tokenizer.**

6 **\* Također sam malo promijenio sintaksu, da omogućim pisanje čistijeg koda**

7 **\* te da podržavam različite vrste podataka (prije je AEC podržavao samo**

8 **\* 32-bitne decimalne brojeve). Novi compiler proizvodi WebAssembly,**

9 **\* standardizirani oblik JavaScriptinog bytecodea. Izvorno je WebAssembly**

10 **\* bio Mozillin standard, ali danas ga podržavaju gotovo sve JavaScriptine**

11 **\* virtualne mašine. Ovdje ciljamo primarno na NodeJS, JavaScriptinu**

12 **\* virtualnu mašinu koju razvija Google i primarno je namijenjena da se**

13 **\* vrti na serverima (no može se pokrenuti i na veoma slabim računalima).**

14 **\*/**

15

16 **// Uvezimo prvo neke funkcije iz JavaScripta, koje će nam trebati...**

17 **Function** daj\_velicinu\_niza**()** **Which** **Returns** **Integer32** **Is** **External;**

18 **Function** kopiraj\_niz\_na\_adresu**(**

19 **Integer32Pointer** adresa**)** **Which** **Returns** **Nothing** **Is** **External;**

20 **Function** printString**(CharacterPointer** str**)** **Which** **Returns** **Nothing** **Is** **External;**

21 **Function** printInteger**(Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

22 **Function** printFloat**(Decimal32** x**)** **Which** **Returns** **Nothing** **Is** **External;**

23 **Function** pocni\_mjerenje\_vremena**()** **Which** **Returns** **Nothing** **Is** **External;**

24 **Function** zavrsi\_mjerenje\_vremena**()** **Which** **Returns** **Nothing** **Is** **External;**

25 **Function** izvijesti\_o\_obrnuto\_poredanim\_nizovima**(**

26 **Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

27 **Function**

28 izvijesti\_o\_poredanim\_nizovima**(Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

29 **Function** izvijesti\_o\_pokretanju\_QuickSorta**(**

30 **Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

31 **Function** izvijesti\_o\_pokretanju\_MergeSorta**(**

32 **Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

33 **Function** izvijesti\_o\_pokretanju\_SelectSorta**(**

34 **Integer32** n**)** **Which** **Returns** **Nothing** **Is** **External;**

35 **Function**

36 izvijesti\_JavaScript\_o\_nedostatku\_memorije**()** **Which** **Returns** **Nothing** **Is** **External;**

37

38 **Integer32** DEBUG **:** **=** **0,** broj\_mjerenja **:** **=** **0;** **// broj\_mjerenja bit će koristan za**

39 **// pokretanje raznih algoritama**

40 **// ovisno o rednom broju mjerenja,**

41 **//što dobro dođe za**

42 **// eksperimentiranje s algoritmima.**

43

44 **// Napravimo sada omotnicu oko WebAssemblerske naredbe "memory.grow"...**

45 **Function** zauzmi\_memorijske\_stranice**(Integer32** broj\_stranica**)** **Which** **Returns**

46 **CharacterPointer**

47 **Does** **{** **// Vitičasta zagrada ovdje je opcionalna, compiler ju ignorira. No,**

48 **// dobro dođe kada pišemo AEC-ovski program u nekom IDE-u koji je**

49 **// primarno namijenjen za C-olike jezike, tada on može razumijeti da**

50 **// je ono između vitičastih zagrada blok naredbi.**

51 **Integer32** nova\_adresa\_u\_stranicama

52 **:** **=** **asm\_i32** **//"asm\_i32" kaže compileru da umetne asemblerski kod, i da**

53 **// pretpostavi da će se nakon njega na sistemskom stogu**

54 **// nalaziti vrijednost tipa "i32". To očito nije točno ako**

55 **// netko prebaci JavaScript virtualnu mašinu u 64-bitni**

56 **// način rada, ali nadam se da to nitko neće napraviti.**

57 **// Vjerojatnost da će JavaScript virtualnoj mašini trebati**

58 **// više nego 4GB RAM-a je zanemariva, a vjerojatnost da će**

59 **// se neki korisni programi srušiti zbog prebacivanja u**

60 **// 64-bitni način rada nije baš zanemariva.**

61 **(**"(memory.grow**\*n***"

62 "**\*t***(local.get 0)**\*n***" **// Prvi (nulti) argument funkcije,**

63 **//"broj\_stranica".**

64 ")**\*n***"**);**

65 **If** nova\_adresa\_u\_stranicama **=** **-1** **Then** **{** **// Ako nema više**

66 **// slobodne memorije...**

67 **Return** **-** **1;**

68 **}**

69 **EndIf;** **// Ni točka-zarez ovdje nije potreban, ali pomaže IDE-ovima da razumiju**

70 **// programski kôd.**

71 **Return** nova\_adresa\_u\_stranicama **\*** **64** **\*** **1024;** **// Na JavaScript Virtualnoj**

72 **// Mašini, jedna stranica**

73 **//(page) iznosi 64 KB.**

74 **}**

75 **EndFunction;**

76

77 **Integer32** velicina\_niza**;**

78 **Integer32Pointer** originalni\_niz**,** pomocni\_niz**;** **// To su globalne varijable,**

79 **// po defaultu su u nuli,**

80 **// dakle "originalni\_niz" i**

81 **//"pomocni\_niz" su na početku**

82 **// programa nulti pokazivači.**

83

84 **Integer32** broj\_obrnuto\_poredanih\_podniza**,** broj\_vec\_poredanih\_podniza**,**

85 broj\_pokretanja\_QuickSorta**,** broj\_pokretanja\_MergeSorta**,**

86 broj\_pokretanja\_SelectSorta**;**

87

88 **// Sad ćemo implementirati neke matematičke funkcije koje će nam trebati.**

89 **// Ne možemo pozvati JavaScriptine matematičke funkcije, jer one su metode**

90 **// singletona "Math", a ne postoji standardizirani način da se zovu**

91 **// metode JavaScriptinih objekata iz WebAssemblyja.**

92 **Decimal32** PRECISION **:** **=** **128;** **// Ovdje možemo balansirati između brzine i**

93 **// preciznosti. Ako smo previše precizni, bit**

94 **//ćemo spori. Ako smo previše neprecizni, lako**

95 **// se može dogoditi da precijenimo koliko se**

96 **// duboko rekurzija smije granati i izazovemo**

97 **// stack overflow.**

98 **Function** prirodni\_logaritam**(Decimal32** x**)** **Which** **Returns** **Decimal32** **Does** **{**

99 **// Prirodni logaritam je integral od 1/x u intervalu od 1 do x,**

100 **// srednjoškolska matematika.**

101 **Decimal32** zbroj **:** **=** **0,** epsilon **:** **=** **(**x **-** **1)** **/** **(5** **\*** PRECISION**),** i **:** **=** **1;**

102 **While(**epsilon **>** **0** **and** i **<** x**)** **or** **(**epsilon **<** **0** **and** i **>** x**)** **Loop** **{**

103 zbroj **+=** epsilon **/** i**;**

104 i **+=** epsilon**;** **// Preuzeo sam naredbe "+=", "-=", "\*=" i "/=" iz C-olikih**

105 **// jezika, smatram da znatno skraćuju neke kodove, a da ih ne**

106 **// čine nečitkima.**

107 **}**

108 **EndWhile;**

109 **Return** zbroj**;**

110 **}**

111 **EndFunction;**

112

113 **Function** Eulerov\_broj\_na\_potenciju**(Decimal32** x**)** **Which** **Returns** **Decimal32** **Does** **{**

114 **// Eulerov Algoritam iz Matematike 2...**

115 **Decimal32** i **:** **=** **0,** y **:** **=** **1,** epsilon **:** **=** x **/** PRECISION**;**

116 **While(**epsilon **>** **0** **and** i **<** x**)** **or** **(**epsilon **<** **0** **and** i **>** x**)** **Loop** **{**

117 y **+=** epsilon **\*** y**;**

118 i **+=** epsilon**;**

119 **}**

120 **EndWhile;**

121 **Return** y**;**

122 **}**

123 **EndFunction;**

124

125 **Function** abs**(Decimal32** x**)** **Which** **Returns** **Decimal32** **Does** **{**

126 **// U svoj sam programski jezik ugradio uvijetni "?:" operator kakav**

127 **// postoji u C-u, C++-u i JavaScriptu. Izgleda malo ružno, ali nekad zna**

128 **// znatno skratiti programske kodove. Odlučio sam implementirati desno**

129 **// asocijativan uvijetni operator, kakav je u C-u, C++-u i JavaScriptu,**

130 **// a ne lijevo asocijativan kakav je u PHP-u i srodnim jezicima.**

131 **// Jednostavno mi ima više smisla da uvijetni operator bude asocijativan**

132 **// na desno nego na lijevo.**

133 **Return(**x **<** **0)** **?** **// Ako je x manji od 0...**

134 **-**x **//...vrati (proglasi rezultatom) -x...**

135 **:** x**;** **// inače, proglasi x rezultatom.**

136 **}**

137 **EndFunction;**

138

139 **Function** ostatak\_pri\_dijeljenju**(Decimal32** x**,**

140 **Decimal32** y**)** **Which** **Returns** **Decimal32** **Does** **{**

141 **If** DEBUG **=** **1** **Then** **{**

142 printString**(**"Zatrazen je ostatak pri dijeljenju od brojeva: "**);**

143 **// Neću upotrebljavati hrvatske znakove u stringovima, jer ću**

144 **// naletjeti na probleme pri pretvorbi u JavaScriptin string.**

145 printFloat**(**x**);**

146 printFloat**(**y**);**

147 printString**(**"Sada ce se program mozda srusiti..."**);**

148 **}**

149 **EndIf;**

150 **If** abs**(**x **/** y**)** **>** Eulerov\_broj\_na\_potenciju**(**prirodni\_logaritam**(2)** **\*** **63)** **Then** **{**

151 **Return** **0;** **// Imate bolju ideju što da se radi u slučaju da količnik**

152 **// ne stane niti u Integer64 (C-ovski "long long")?**

153 **}**

154 **EndIf;**

155 **Return** x **-** y **\*Integer64(**x **/** y**);** **// Ako napišem "Integer32",**

156 **// riskiram da će JavaScript**

157 **// virtualna mašina prekinuti**

158 **// izvođenje programa jer je**

159 **// broj "x/y" izvan intervala**

160 **// koji 32-bitni cijeli brojevi**

161 **// mogu prikazati (od oko dvije**

162 **// milijarde u pozitivno i**

163 **// negativno).**

164 **}**

165 **EndFunction;**

166

167 **Function** pow**(Decimal32** x**,** **Decimal32** y**)** **Which** **Returns** **Decimal32** **Does** **{**

168 **Decimal32** result

169 **:** **=** Eulerov\_broj\_na\_potenciju**(**prirodni\_logaritam**(**abs**(**x**))** **\*** y**);**

170 **Return** x **=**

171 **0** **?** **0** **:** ostatak\_pri\_dijeljenju**(**x**,** **2)** **=** **1** **and** x **<** **0** **?** **-**result **:** result**;**

172 **}**

173 **EndFunction;**

174

175 **// I sada krećemo pisati taj hibridni algoritam razvrstavanja...**

176 **Function** hybrid\_sort**(Integer32** donja\_granica**,** **Integer32** gornja\_granica**,**

177 **Integer32** dubina\_rekurzije**)** **Which** **Returns** **Nothing** **Does** **{**

178 **If** gornja\_granica **-** donja\_granica **<** **2** **Then** **{** **// Ako je niz duljine manje od**

179 **// 2 (0 ili 1), znači da je već**

180 **// poredan, pa prekidamo**

181 **// izvođenje ovog potprograma.**

182 **Return;**

183 **}**

184 **ElseIf** gornja\_granica **-** donja\_granica **=** **2** **Then** **{** **// Najčesći slučaj,**

185 **// vrijedi ga posebno**

186 **// obraditi jer time**

187 **// možemo znatno ubrzati**

188 **// program.**

189 **If** **ValueAt(**originalni\_niz **+** donja\_granica**)** **>**

190 **ValueAt(**originalni\_niz **+** donja\_granica **+** **1)** **Then** **{**

191 **Integer32** pomocna\_varijabla\_za\_zamijenu

192 **:** **=** **ValueAt(**originalni\_niz **+** donja\_granica**);**

193 **ValueAt(**originalni\_niz **+** donja\_granica**)**

194 **:** **=** **ValueAt(**originalni\_niz **+** donja\_granica **+** **1);**

195 **ValueAt(**originalni\_niz **+** donja\_granica **+** **1)**

196 **:** **=** pomocna\_varijabla\_za\_zamijenu**;**

197 **}**

198 **EndIf;**

199 **Return;**

200 **}**

201 **ElseIf** gornja\_granica **-**

202 donja\_granica**<8** **or** **asm\_i32(**"(global.get $stack\_pointer)"**)>** **4** **\*** **1024** **-**

203 **73** **Then** **{**

204 **// Za male je nizove**

205 **// SelectionSort brži**

206 **// i od MergeSorta i**

207 **// QuickSorta.**

208 **// Također, kako nije**

209 **// rekurzivan, može se**

210 **// koristiti i kad posve**

211 **// potrošimo memoriju**

212 **// na sistemskom stogu**

213 **//(na JavaScript**

214 **// Virtualnoj Mašini to**

215 **// jest ne više nego**

216 **// 4 KB, kako bijaše**

217 **// u doba Netscapea 2,**

218 **// godine 1996, tako i**

219 **// danas).**

220 broj\_pokretanja\_SelectSorta**:**

221 **=** broj\_pokretanja\_SelectSorta **+** **1;**

222 **Integer32** i **:** **=** donja\_granica**;**

223 **While** i **<** gornja\_granica **Loop** **{**

224 **Integer32** gdje\_je\_minimum **:** **=** i**;**

225 **Integer32** j **:** **=** i **+** **1;**

226 **While** j **<** gornja\_granica **Loop** **{**

227 **If** **ValueAt(**originalni\_niz **+** gdje\_je\_minimum**)** **>**

228 **ValueAt(**originalni\_niz **+** j**)** **Then** **{**

229 gdje\_je\_minimum**:**

230 **=** j**;**

231 **}**

232 **EndIf;**

233 j **+=** **1;**

234 **}**

235 **EndWhile;**

236 **Integer32** pomocna\_varijabla\_za\_zamijenu **:** **=** **ValueAt(**originalni\_niz **+** i**);**

237 **ValueAt(**originalni\_niz **+** i**)** **:** **=** **ValueAt(**originalni\_niz **+** gdje\_je\_minimum**);**

238 **ValueAt(**originalni\_niz **+** gdje\_je\_minimum**)**

239 **:** **=** pomocna\_varijabla\_za\_zamijenu**;**

240 i **+=** **1;**

241 **}**

242 **EndWhile;**

243 **Return;**

244 **}**

245 **EndIf;**

246 **Decimal32** razvrstanost **:** **=** **0;**

247 **Integer32** i **:** **=** donja\_granica**,** je\_li\_niz\_vec\_poredan **:** **=** **1;**

248 **While** i **<** gornja\_granica **-** **1** **Loop** **{**

249 razvrstanost **+=**

250 **(ValueAt(**originalni\_niz **+** i**)** **<** **ValueAt(**originalni\_niz **+** i **+** **1)** **or**

251 **ValueAt(**originalni\_niz **+** i**)** **=** **ValueAt(**originalni\_niz **+** i **+** **1));**

252 **If** **ValueAt(**originalni\_niz **+** i**)** **>** **ValueAt(**originalni\_niz **+** i **+** **1)** **Then** **{**

253 je\_li\_niz\_vec\_poredan**:**

254 **=** **0;**

255 **}**

256 **EndIf;**

257 i **+=** **1;**

258 **}**

259 **EndWhile;**

260 razvrstanost**:**

261 **=** razvrstanost **/** **((**gornja\_granica **-** donja\_granica **-** **1)** **/** **2.)** **-** **1;**

262 **// Provjeri je li sve u redu, i, ako nije, obavijesti.**

263 **If** abs**(**razvrstanost**)** **>** **1** **Then** **{**

264 **// To ne smije biti...**

265 printString**(**"Apsolutna vrijednost razvrstanosti je veca od 1!"**);**

266 printString**(**"Relevantni dio niza iznosi:"**);** **// Da se ne moram baktati s**

267 **// debuggerima za JavaScript**

268 **// virtualnu mašinu ako dođe**

269 **// do problema, lakše mi**

270 **// je ispisati brojeve u**

271 **// programu nego tražiti**

272 **// kako narediti**

273 **// debuggeru da ih**

274 **// ispiše.**

275 i**:**

276 **=** donja\_granica**;**

277 **While** i **<** gornja\_granica **Loop** **{**

278 printInteger**(ValueAt(**originalni\_niz **+** i**));**

279 i **+=** **1;**

280 **}**

281 **EndWhile;**

282 printString**(**"Kraj relevantnog dijela niza!"**);**

283 **}**

284 **EndIf;**

285 **If** je\_li\_niz\_vec\_poredan **and** **not(**razvrstanost **=** **1** **or** razvrstanost **>** **1)** **Then** **{**

286 **// Opet ne smije biti...**

287 printString**(**"Niz je poredan, a razvrstanost nije 1."**);**

288 printString**(**"Relevantni dio niza iznosi:"**);**

289 i**:**

290 **=** donja\_granica**;**

291 **While** i **<** gornja\_granica **Loop** **{**

292 printInteger**(ValueAt(**originalni\_niz **+** i**));**

293 i **+=** **1;**

294 **}**

295 **EndWhile;**

296 printString**(**"Kraj relevantnog dijela niza!"**);**

297 **}**

298 **EndIf;**

299 **If** **not(**je\_li\_niz\_vec\_poredan**)** **and** **(**razvrstanost **=** **1** **or** razvrstanost **>** **1)**

300 **Then** **{**

301 **// Open ne smije biti...**

302 printString**(**"Razvrstanost je 1, a niz nije poredan!"**);**

303 printString**(**"Relevantni dio niza iznosi:"**);**

304 i**:**

305 **=** donja\_granica**;**

306 **While** i **<** gornja\_granica **Loop** **{**

307 printInteger**(ValueAt(**originalni\_niz **+** i**));**

308 i **+=** **1;**

309 **}**

310 **EndWhile;**

311 printString**(**"Kraj relevantnog dijela niza!"**);**

312 **}**

313 **EndIf;**

314 **// Idemo dalje...**

315 **Decimal32** razvrstanost\_na\_potenciju**[8]** **:** **=** **{1};** **// Formula će se brže**

316 **// izračunati ako ne**

317 **// pozivamo "pow" gdje**

318 **// ne treba (kad je**

319 **// eksponent prirodan**

320 **// broj).**

321i**:**

322 **=** **1;**

323 **While** i **<** **8** **Loop** **{**

324 razvrstanost\_na\_potenciju**[**i**]**

325 **:** **=** razvrstanost\_na\_potenciju**[**i **-** **1]** **\*** razvrstanost**;**

326 i **+=** **1;**

327 **}**

328 **EndWhile;**

329 **// Formula koju je ispisao genetski algoritam za predviđanje koliko će**

330 **// usporedbi QuickSort napraviti:**

331 **// https://github.com/FlatAssembler/ArithmeticExpressionCompiler/tree/master/QuickSort/Genetic\_algorithm\_for\_deriving\_the\_formula**

332 **Decimal32** polinom\_pod\_apsolutnom

333 **:** **=** **2.38854** **\*** razvrstanost\_na\_potenciju**[7]** **-**

334 **0.284258** **\*** razvrstanost\_na\_potenciju**[6]** **-**

335 **1.87104** **\*** razvrstanost\_na\_potenciju**[5]** **+**

336 **0.372637** **\*** razvrstanost\_na\_potenciju**[4]** **+**

337 **0.167242** **\*** razvrstanost\_na\_potenciju**[3]** **-**

338 **0.0884977** **\*** razvrstanost\_na\_potenciju**[2]** **+** **0.315119** **\*** razvrstanost**;**

339 **Decimal32** Eulerov\_broj\_na\_koju\_potenciju

340 **:** **=** **(**prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)** **+**

341 prirodni\_logaritam**(**

342 prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)))** **\***

343 **1.05** **+**

344 **(**prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)** **-**

345 prirodni\_logaritam**(**

346 prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**))** **-**

347 prirodni\_logaritam**(2))** **\***

348 **0.9163** **\*** abs**(**polinom\_pod\_apsolutnom**);**

349 **Decimal32** koliko\_usporedbi\_ocekujemo\_od\_QuickSorta

350 **:** **=** Eulerov\_broj\_na\_potenciju**(**Eulerov\_broj\_na\_koju\_potenciju**);**

351 **Decimal32** koliko\_usporedbi\_ocekujemo\_od\_MergeSorta

352 **:** **=** **(mod(**broj\_mjerenja**,** **2)** **+** **1)** **\***

353 **(**gornja\_granica **-**

354 donja\_granica**)** **\*** **// Nisam siguran treba li ovdje**

355 **// pisati "2 \* (gornja\_granica...".**

356 **// S jedne strane, MergeSort radi**

357 **// dvije petlje, jedna za spajanje**

358 **// dijelova originalnog niza u**

359 **// pomoćni niz, a druga za kopiranje**

360 **// pomoćnog niza u originalni. S**

361 **// druge strane, iz mjerenja se čini**

362 **// da je cjelokupni algoritam brži**

363 **// ako se ne množi s 2. Zato ćemo**

364 **// nekada množiti s dva, a nekada ne,**

365 **// pa ćemo preciznijim mjerenjima**

366 **// vidjeti što je bolje.**

367 prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)** **/**

368 prirodni\_logaritam**(2);**

369 **// I sada kreće grananje na temelju izračunatog...**

370 **If** razvrstanost **=** **1** **or** razvrstanost **>** **1** **Then** **{**

371 broj\_vec\_poredanih\_podniza **+=** **1;**

372 **Return;**

373 **}**

374 **ElseIf** razvrstanost **=** **-1** **or** razvrstanost **<** **-1** **Then** **{**

375 broj\_obrnuto\_poredanih\_podniza **+=** **1;**

376 **Integer32** i **:** **=** donja\_granica**;**

377 **Integer32** j **:** **=** gornja\_granica **-** **1;**

378 **While** i **<** gornja\_granica **Loop** **{**

379 **ValueAt(**pomocni\_niz **+** i**)** **:** **=** **ValueAt(**originalni\_niz **+** j**);**

380 j **-=** **1;**

381 i **+=** **1;**

382 **}**

383 **EndWhile;**

384 i**:**

385 **=** donja\_granica**;**

386 **While** i **<** gornja\_granica **Loop** **{**

387 **ValueAt(**originalni\_niz **+** i**)** **:** **=** **ValueAt(**pomocni\_niz **+** i**);**

388 i **+=** **1;**

389 **}**

390 **EndWhile;**

391 **Return;**

392 **}**

393 **ElseIf(**koliko\_usporedbi\_ocekujemo\_od\_MergeSorta **<**

394 koliko\_usporedbi\_ocekujemo\_od\_QuickSorta **or**

395 dubina\_rekurzije **>** pow**(2,** **18** **-** prirodni\_logaritam**(**velicina\_niza**)** **/**

396 prirodni\_logaritam**(2))**

397 **// JavaScriptina virtualna mašina ima**

398 **// 4KB memorije na sistemskom stogu,**

399 **// i alociranje više heap memorije**

400 **// ne mijenja tu nesretnu činjenicu.**

401 **// Ne znam kako Emscripten (modificirana**

402 **// verzija CLANG-a koja compilira**

403 **// C++ u WebAssembly) to rješava.**

404 **)** **and**

405 **not(**gornja\_granica **-** donja\_granica **=**

406 velicina\_niza **and** **not(mod(**broj\_mjerenja**,** **3)))**

407 **// Izgleda da je, iz nekog razloga, program brži ako se QuickSort**

408 **// pokrene barem jednom, no probajmo raditi preciznija mjerenja.**

409 **Then** **{**

410 **// MergeSort algoritam (približno poredani podnizovi,**

411 **// za koje je MergeSort efikasniji od QuickSorta,**

412 **// a moj ga program također koristi kada ima još**

413 **// malo mjesta na sistemskom stogu, pa QuickSort**

414 **// nije opcija)...**

415 broj\_pokretanja\_MergeSorta **+=** **1;**

416 **Integer32** sredina\_niza **:** **=** **(**gornja\_granica **+** donja\_granica**)** **/** **2;**

417 **// Prvo, rastavi niz na koji pokazuje pokazivač "originalni\_niz"**

418 **// na niz od originalni\_niz+donja\_granica do**

419 **// originalni\_niz+sredina\_niza i niz od**

420 **// originalni\_niz+sredina\_niza do**

421 **// originalni\_niz+gornja\_granica,**

422 **// i poredaj ta dva niza.**

423 hybrid\_sort**(**donja\_granica**,** sredina\_niza**,** dubina\_rekurzije **+** **1);**

424 hybrid\_sort**(**sredina\_niza**,** gornja\_granica**,** dubina\_rekurzije **+** **1);**

425 **// Spajanje nizova originalni\_niz[donja\_granica..sredina\_niza]**

426 **// i originalni\_niz[sredina\_niza..gornja\_granica] u jedan niz...**

427 **Integer32** i **:** **=** donja\_granica**;**

428 **Integer32** gdje\_smo\_u\_prvom\_nizu **:** **=** donja\_granica**;**

429 **Integer32** gdje\_smo\_u\_drugom\_nizu **:** **=** sredina\_niza**;**

430 **While** i **<** gornja\_granica **Loop** **{**

431 **If(**gdje\_smo\_u\_prvom\_nizu **=**

432 sredina\_niza **or**

433 **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_drugom\_nizu**)** **<**

434 **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_prvom\_nizu**))** **and**

435 gdje\_smo\_u\_drugom\_nizu **<** gornja\_granica **Then** **{**

436 **ValueAt(**pomocni\_niz **+** i**)**

437 **:** **=** **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_drugom\_nizu**);**

438 gdje\_smo\_u\_drugom\_nizu **+=** **1;**

439 **}**

440 **Else** **{**

441 **ValueAt(**pomocni\_niz **+** i**)**

442 **:** **=** **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_prvom\_nizu**);**

443 gdje\_smo\_u\_prvom\_nizu **+=** **1;**

444 **}**

445 **EndIf;**

446 i **+=** **1;**

447 **}**

448 **EndWhile;**

449 i**:**

450 **=** donja\_granica**;**

451 **While** i **<** gornja\_granica **Loop** **{**

452 **ValueAt(**originalni\_niz **+** i**)** **:** **=** **ValueAt(**pomocni\_niz **+** i**);**

453 i **+=** **1;**

454 **}**

455 **EndWhile;**

456 **Return;**

457 **}**

458 **Else** **{** **// QuickSort algoritam (nasumično ispremještani podnizovi)...**

459 broj\_pokretanja\_QuickSorta **+=** **1;**

460 **// Daljnji kod je približno prepisan s**

461 **// https://www.geeksforgeeks.org/quick-sort/**

462 **// Iskreno, ne razumijem ni ja točno kako funkcionira.**

463 **// On navodno preuređuje niz tako da svi elementi koji su manji**

464 **// od onog koji je bio prvi (pivot) dođu prije njega, a ostali**

465 **// poslije njega.**

466 **Integer32** pivot **:** **=** **ValueAt(**originalni\_niz **+** gornja\_granica **-** **1);**

467 **Integer32** i **:** **=** donja\_granica **-** **1;**

468 **Integer32** j **:** **=** donja\_granica**;**

469 **Integer32** pomocna\_varijabla\_za\_zamijenu**;**

470 **While** j **<** gornja\_granica **-** **1** **Loop** **{**

471 **If** **ValueAt(**originalni\_niz **+** j**)** **<** pivot **Then** **{**

472 i **+=** **1;**

473 pomocna\_varijabla\_za\_zamijenu**:**

474 **=** **ValueAt(**originalni\_niz **+** i**);**

475 **ValueAt(**originalni\_niz **+** i**)** **:** **=** **ValueAt(**originalni\_niz **+** j**);**

476 **ValueAt(**originalni\_niz **+** j**)** **:** **=** pomocna\_varijabla\_za\_zamijenu**;**

477 **}**

478 **EndIf;**

479 j **+=** **1;**

480 **}**

481 **EndWhile;**

482 pomocna\_varijabla\_za\_zamijenu**:**

483 **=** **ValueAt(**originalni\_niz **+** i **+** **1);**

484 **ValueAt(**originalni\_niz **+** i **+** **1)**

485 **:** **=** **ValueAt(**originalni\_niz **+** gornja\_granica **-** **1);**

486 **ValueAt(**originalni\_niz **+** gornja\_granica **-** **1)**

487 **:** **=** pomocna\_varijabla\_za\_zamijenu**;**

488 **Integer32** gdje\_je\_pivot **:** **=** i **+** **1;**

489 hybrid\_sort**(**donja\_granica**,** gdje\_je\_pivot**,** dubina\_rekurzije **+** **1);**

490 hybrid\_sort**(**gdje\_je\_pivot**,** gornja\_granica**,** dubina\_rekurzije **+** **1);**

491 **Return;**

492 **}**

493 **EndIf;**

494 **// Ovdje tok programa ne smije doći.**

495 printString**(**"Izgleda da compiler nije ispravno "

496 "preveo kontrolne strukture!"**);**

497 **}**

498 **EndFunction;**

499

500 **// Ovo je funkcija koju će pozvati JavaScript...**

501 **Function** pocetna\_AEC\_funkcija**()** **Which** **Returns** **Nothing** **Does** **{**

502 **If** originalni\_niz **=** **-1** **or** pomocni\_niz **=** **-1** **Then** **{**

503 **Return;** **// Ako JavaScript nastavlja pokretati ovaj program**

504 **// unatoč nedostatku memorije, neka onda on ne radi ništa.**

505 **}**

506 **EndIf;**

507 **// Testiraj matematičke funkcije...**

508 **If** abs**(**pow**(3,** **3)** **-** **27)** **>** **2** **Then** **{** **// Da, one su jako neprecizne, ali zato**

509 **// jako brze.**

510 printString**(**"Izgleda da matematicke funkcije ne funkcioniraju dobro."**);**

511 printString**(**"pow(3, 3) ="**);**

512 printFloat**(**pow**(3,** **3));**

513 **}**

514 **EndIf;**

515 **// Doznaj veličinu niza iz JavaScripta...**

516 **Integer32** prijasnja\_velicina\_niza **:** **=** velicina\_niza**;**

517 velicina\_niza**:**

518 **=** daj\_velicinu\_niza**();**

519 **// Ako je potrebno, zauzmi još memorije...**

520 **If** velicina\_niza **/** **(64** **\*** **1024** **/** **4)** **+**

521 **not(not(mod(**velicina\_niza**,** **64** **\*** **1024** **/** **4)))** **>**

522 prijasnja\_velicina\_niza **/** **(64** **\*** **1024** **/** **4)** **+**

523 **not(not(mod(**prijasnja\_velicina\_niza**,** **64** **\*** **1024** **/** **4)))** **or**

524 prijasnja\_velicina\_niza **=** **0** **Then** **{**

525 originalni\_niz**:**

526 **=** zauzmi\_memorijske\_stranice**(4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

527 **not(not(mod(**velicina\_niza**,** **64** **\*** **1024** **/** **4))));**

528 pomocni\_niz**:**

529 **=** zauzmi\_memorijske\_stranice**(4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

530 **not(not(mod(**velicina\_niza**,** **64** **\*** **1024** **/** **4))));**

531 **If** originalni\_niz **=** **-1** **or** pomocni\_niz **=** **-1** **Then** **{**

532 printString**(**"Nema dovoljno memorije za nastavak programa!?"**);**

533 izvijesti\_JavaScript\_o\_nedostatku\_memorije**();**

534 **Return;** **// Prekini izvršavanje ovog programa.**

535 **}**

536 **EndIf;**

537 **}**

538 **EndIf;**

539 **// Sada zatraži od JavaScripta da kopira niz koji treba poredati**

540 **// na memorijski prostor koji si (prethodno ili sada) zauzeo.**

541 kopiraj\_niz\_na\_adresu**(**originalni\_niz**);**

542 **// I sada ga kreni razvrstavati i mjeriti koliko ti treba vremena.**

543broj\_obrnuto\_poredanih\_podniza**:**

544 **=** broj\_vec\_poredanih\_podniza **:** **=** broj\_pokretanja\_QuickSorta

545 **:** **=** broj\_pokretanja\_MergeSorta **:** **=** broj\_pokretanja\_SelectSorta

546 **:** **=** **0;** **// Nisam mogao odoljeti da u svoj programski jezik ne dodam**

547 **// ulančano pridruživanje iz C-a, C++-a i JavaScripta (da možemo**

548 **// više varijabli postaviti na neku vrijednost u jednoj naredbi).**

549 broj\_mjerenja **+=** **1;**

550 pocni\_mjerenje\_vremena**();**

551 hybrid\_sort**(0,** velicina\_niza**,** **0);**

552 zavrsi\_mjerenje\_vremena**();**

553 **// Kad završi mjerenje vremena (koje se vrtilo u JavaScriptu),**

554 **// obavijesti JavaScript o onome što si ti izmjerio.**

555 izvijesti\_o\_obrnuto\_poredanim\_nizovima**(**broj\_obrnuto\_poredanih\_podniza**);**

556 izvijesti\_o\_poredanim\_nizovima**(**broj\_vec\_poredanih\_podniza**);**

557 izvijesti\_o\_pokretanju\_QuickSorta**(**broj\_pokretanja\_QuickSorta**);**

558 izvijesti\_o\_pokretanju\_MergeSorta**(**broj\_pokretanja\_MergeSorta**);**

559 izvijesti\_o\_pokretanju\_SelectSorta**(**broj\_pokretanja\_SelectSorta**);**

560 **// Napravi neki osnovni sanity-check, je li niz uistinu poredan?**

561 **Integer32** i **:** **=** **0;**

562 **While** i **<** velicina\_niza **-** **1** **Loop** **{**

563 **If** **ValueAt(**originalni\_niz **+** i**)** **>** **ValueAt(**originalni\_niz **+** i **+** **1)** **Then** **{**

564 printString**(**"Niz nije poredan!"**);**

565 **Return;** **// Nemoj to ispisati više puta, nego prekini program čim**

566 **// si uočio prvu nepodudarnost.**

567 **}**

568 **EndIf;**

569 i **+=** **1;**

570 **}**

571 **EndWhile;**

572 **}**

573 **EndFunction;**

574