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**(Integer32Pointer** adresa**)**

19 **Which** **Returns** **Nothing** **Is** **External;**

20 **Function** printString**(CharacterPointer** str**)**

21 **Which** **Returns** **Nothing** **Is** **External;**

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

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

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

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

26 **Function** izvijesti\_o\_obrnuto\_poredanim\_nizovima**(Integer32** n**)**

27 **Which** **Returns** **Nothing** **Is** **External;**

28 **Function** izvijesti\_o\_poredanim\_nizovima**(Integer32** n**)**

29 **Which** **Returns** **Nothing** **Is** **External;**

30 **Function** izvijesti\_o\_pokretanju\_QuickSorta**(Integer32** n**)**

31 **Which** **Returns** **Nothing** **Is** **External;**

32 **Function** izvijesti\_o\_pokretanju\_MergeSorta**(Integer32** n**)**

33 **Which** **Returns** **Nothing** **Is** **External;**

34 **Function** izvijesti\_o\_pokretanju\_SelectSorta**(Integer32** n**)**

35 **Which** **Returns** **Nothing** **Is** **External;**

36 **Function** izvijesti\_JavaScript\_o\_nedostatku\_memorije**()**

37 **Which** **Returns** **Nothing** **Is** **External;**

38

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

40 **//pokretanje raznih algoritama**

41 **//ovisno o rednom broju mjerenja**

42 **//što dobro dođe za**

43 **//eksperimentiranje s algoritmima.**

44

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

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

47 **CharacterPointer** **Does**

48 **Integer32** nova\_adresa\_u\_stranicama **:=**

49 **asm\_i32** **//"asm\_i32" kaže compileru da umetne asemblerski kod, i da**

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

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

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

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

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

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

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

57 **//64-bitni način rada nije baš zanemariva.**

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

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

60 **//"broj\_stranica".**

61 ")**\*n***"**);**

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

63 **//slobodne memorije...**

64 **Return** **-1;**

65 **EndIf**

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

67 **//Mašini, jedna stranica**

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

69 **EndFunction**

70

71 **Integer32** velicina\_niza**;**

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

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

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

75 **//"pomocni\_niz" su na početku**

76 **//programa nulti pokazivači.**

77

78 **Integer32** broj\_obrnuto\_poredanih\_podniza**,**

79 broj\_vec\_poredanih\_podniza**,**

80 broj\_pokretanja\_QuickSorta**,**

81 broj\_pokretanja\_MergeSorta**,**

82 broj\_pokretanja\_SelectSorta**;**

83

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

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

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

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

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

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

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

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

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

93 **//stack overflow.**

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

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

96 **//srednjoškolska matematika.**

97 **Decimal32** zbroj **:=** **0,**

98 epsilon **:=** **(**x **-** **1)** **/** **(5** **\*** PRECISION**),**

99 i **:=** **1;**

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

101 zbroj**:=** zbroj **+** epsilon **/** i**;**

102 i **:=** i **+** epsilon**;**

103 **EndWhile**

104 **Return** zbroj**;**

105 **EndFunction**

106

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

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

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

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

111 y **:=** y **+** epsilon **\*** y**;**

112 i **:=** i **+** epsilon**;**

113 **EndWhile**

114 **Return** y**;**

115 **EndFunction**

116

117 **Function** abs**(Decimal32** x**)** **Which** **Returns** **Decimal32** **Does**

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

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

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

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

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

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

124 **//na desno nego na lijevo.**

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

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

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

128 **EndFunction**

129

130 **Function** ostatak\_pri\_dijeljenju**(Decimal32** x**,** **Decimal32** y**)**

131 **Which** **Returns** **Decimal32** **Does**

132 **If** DEBUG **=** **1** **Then**

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

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

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

136 printFloat**(**x**);**

137 printFloat**(**y**);**

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

139 **EndIf**

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

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

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

143 **EndIf**

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

145 **//riskiram da će JavaScript**

146 **//virtualna mašina prekinuti**

147 **//izvođenje programa jer je**

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

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

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

151 **//milijarde u pozitivno i**

152 **//negativno).**

153 **EndFunction**

154

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

156 **Decimal32** result **:=** Eulerov\_broj\_na\_potenciju**(**

157 prirodni\_logaritam**(**abs**(**x**))** **\*** y**);**

158 **Return** x **=** **0?**

159 **0**

160 **:** ostatak\_pri\_dijeljenju**(**x**,** **2)** **=** **1** **and** x **<** **0** **?**

161 **-**result

162 **:** result**;**

163 **EndFunction**

164

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

166 **Function** hybrid\_sort**(Integer32** donja\_granica**,**

167 **Integer32** gornja\_granica**,**

168 **Integer32** dubina\_rekurzije**)**

169 **Which** **Returns** **Nothing** **Does**

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

171 **//2 (0 ili 1), znači da je već**

172 **//poredan, pa prekidamo**

173 **//izvođenje ovog potprograma.**

174 **Return;**

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

176 **//vrijedi ga posebno**

177 **//obraditi jer time**

178 **//možemo znatno ubrzati**

179 **//program.**

180 **If**

181 **ValueAt(**originalni\_niz **+** donja\_granica**)** **>**

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

183 **Then**

184 **Integer32** pomocna\_varijabla\_za\_zamijenu **:=**

185 **ValueAt(**originalni\_niz **+** donja\_granica**);**

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

187 **ValueAt(**originalni\_niz **+** donja\_granica **+** **1);**

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

189 pomocna\_varijabla\_za\_zamijenu**;**

190 **EndIf**

191 **Return;**

192 **ElseIf** gornja\_granica **-** donja\_granica **<** **8** **or**

193 **asm\_i32(**"(global.get $stack\_pointer)"**)** **>** **4** **\*** **1024** **-** **73** **Then**

194 **//Za male je nizove**

195 **//SelectionSort brži**

196 **//i od MergeSorta i**

197 **//QuickSorta.**

198 **//Također, kako nije**

199 **//rekurzivan, može se**

200 **//koristiti i kad posve**

201 **//potrošimo memoriju**

202 **//na sistemskom stogu**

203 **//(na JavaScript**

204 **//Virtualnoj Mašini to**

205 **//jest ne više nego**

206 **//4 KB, kako bijaše**

207 **//u doba Netscapea 2,**

208 **//godine 1996, tako i**

209 **//danas).**

210 broj\_pokretanja\_SelectSorta **:=** broj\_pokretanja\_SelectSorta **+** **1;**

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

212 **While** i **<** gornja\_granica **Loop**

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

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

215 **While** j **<** gornja\_granica **Loop**

216 **If**

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

218 **ValueAt(**originalni\_niz **+** j**)**

219 **Then**

220 gdje\_je\_minimum **:=** j**;**

221 **EndIf**

222 j **:=** j **+** **1;**

223 **EndWhile**

224 **Integer32** pomocna\_varijabla\_za\_zamijenu **:=**

225 **ValueAt(**originalni\_niz **+** i**);**

226 **ValueAt(**originalni\_niz **+** i**)** **:=**

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

228 **ValueAt(**originalni\_niz **+** gdje\_je\_minimum**)** **:=**

229 pomocna\_varijabla\_za\_zamijenu**;**

230 i **:=** i **+** **1;**

231 **EndWhile**

232 **Return;**

233 **EndIf**

234 **Decimal32** razvrstanost **:=** **0;**

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

236 **While** i **<** gornja\_granica **-** **1** **Loop**

237 razvrstanost **:=** razvrstanost **+**

238 **(ValueAt(**originalni\_niz **+** i**)** **<** **ValueAt(**originalni\_niz **+** i **+** **1)**

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

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

241 je\_li\_niz\_vec\_poredan **:=** **0;**

242 **EndIf**

243 i **:=** i **+** **1;**

244 **EndWhile**

245 razvrstanost **:=** razvrstanost **/**

246 **(** **(**gornja\_granica **-** donja\_granica **-** **1)** **/** **2.)** **-** **1;**

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

248 **If** abs**(**razvrstanost**)** **>** **1** **Then**

249 **//To ne smije biti...**

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

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

252 **//debuggerima za JavaScript**

253 **//virtualnu mašinu ako dođe**

254 **//do problema, lakše mi**

255 **//je ispisati brojeve u**

256 **//programu nego tražiti**

257 **//kako narediti**

258 **//debuggeru da ih**

259 **//ispiše.**

260 i **:=** donja\_granica**;**

261 **While** i **<** gornja\_granica **Loop**

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

263 i **:=** i **+** **1;**

264 **EndWhile**

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

266 **EndIf**

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

268 **//Opet ne smije biti...**

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

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

271 i **:=** donja\_granica**;**

272 **While** i **<** gornja\_granica **Loop**

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

274 i **:=** i **+** **1;**

275 **EndWhile**

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

277 **EndIf**

278 **If**

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

280 **Then**

281 **//Open ne smije biti...**

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

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

284 i **:=** donja\_granica**;**

285 **While** i **<** gornja\_granica **Loop**

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

287 i **:=** i **+** **1;**

288 **EndWhile**

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

290 **EndIf**

291 **//Idemo dalje...**

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

293 **//izračunati ako ne**

294 **//pozivamo "pow" gdje**

295 **//ne treba (kad je**

296 **//eksponent prirodan**

297 **//broj).**

298 i **:=** **1;**

299 **While** i **<** **8** **Loop**

300 razvrstanost\_na\_potenciju**[**i**]** **:=**

301 razvrstanost\_na\_potenciju**[**i **-** **1]** **\*** razvrstanost**;**

302 i **:=** i **+** **1;**

303 **EndWhile**

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

305 **//usporedbi QuickSort napraviti:**

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

307 **Decimal32** polinom\_pod\_apsolutnom **:=**

308 **2.38854** **\*** razvrstanost\_na\_potenciju**[7]**

309 **-** **0.284258** **\*** razvrstanost\_na\_potenciju**[6]**

310 **-** **1.87104** **\*** razvrstanost\_na\_potenciju**[5]**

311 **+** **0.372637** **\*** razvrstanost\_na\_potenciju**[4]**

312 **+** **0.167242** **\*** razvrstanost\_na\_potenciju**[3]**

313 **-** **0.0884977** **\*** razvrstanost\_na\_potenciju**[2]**

314 **+** **0.315119** **\*** razvrstanost**;**

315 **Decimal32** Eulerov\_broj\_na\_koju\_potenciju **:=**

316 **(**prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)** **+**

317 prirodni\_logaritam**(**prirodni\_logaritam**(**gornja\_granica

318 **-** donja\_granica**)))** **\*** **1.05**

319 **+** **(**prirodni\_logaritam**(**gornja\_granica **-** donja\_granica**)** **-**

320 prirodni\_logaritam**(**prirodni\_logaritam**(**gornja\_granica **-**

321 donja\_granica**))** **-** prirodni\_logaritam**(2))**

322 **\*** **0.9163** **\*** abs**(**polinom\_pod\_apsolutnom**);**

323 **Decimal32** koliko\_usporedbi\_ocekujemo\_od\_QuickSorta **:=**

324 Eulerov\_broj\_na\_potenciju**(**Eulerov\_broj\_na\_koju\_potenciju**);**

325 **Decimal32** koliko\_usporedbi\_ocekujemo\_od\_MergeSorta **:=**

326 **(mod(**broj\_mjerenja**,** **2)** **+** **1)** **\***

327 **(**gornja\_granica **-** donja\_granica**)** **\*** **//Nisam siguran treba li ovdje**

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

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

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

331 **//dijelova originalnog niza u**

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

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

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

335 **//da je cjelokupni algoritam brži**

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

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

338 **//pa ćemo preciznijim mjerenjima**

339 **//vidjeti što je bolje.**

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

341 **/** prirodni\_logaritam**(2);**

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

343 **If** razvrstanost **=** **1** **or** razvrstanost **>** **1** **Then**

344 broj\_vec\_poredanih\_podniza **:=** broj\_vec\_poredanih\_podniza **+** **1;**

345 **Return;**

346 **ElseIf** razvrstanost **=** **-1** **or** razvrstanost **<** **-1** **Then**

347 broj\_obrnuto\_poredanih\_podniza **:=** broj\_obrnuto\_poredanih\_podniza **+** **1;**

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

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

350 **While** i **<** gornja\_granica **Loop**

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

352 j **:=** j **-** **1;**

353 i **:=** i **+** **1;**

354 **EndWhile**

355 i **:=** donja\_granica**;**

356 **While** i **<** gornja\_granica **Loop**

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

358 i **:=** i **+** **1;**

359 **EndWhile**

360 **Return;**

361 **ElseIf**

362 **(**koliko\_usporedbi\_ocekujemo\_od\_MergeSorta **<**

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

364 dubina\_rekurzije **>** pow**(2,**

365 **18** **-** prirodni\_logaritam**(**velicina\_niza**)** **/** prirodni\_logaritam**(2))**

366 **//JavaScriptina virtualna mašina ima**

367 **//4KB memorije na sistemskom stogu,**

368 **//i alociranje više heap memorije**

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

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

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

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

373 **)** **and** **not(**gornja\_granica **-** donja\_granica **=** velicina\_niza

374 **and** **not(mod(**broj\_mjerenja**,** **3)))**

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

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

377 **Then**

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

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

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

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

382 **//nije opcija)...**

383 broj\_pokretanja\_MergeSorta **:=** broj\_pokretanja\_MergeSorta **+** **1;**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

398 **While** i **<** gornja\_granica **Loop**

399 **If** **(**gdje\_smo\_u\_prvom\_nizu **=** sredina\_niza **or**

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

401 **<** **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_prvom\_nizu**))**

402 **and** gdje\_smo\_u\_drugom\_nizu **<** gornja\_granica **Then**

403 **ValueAt(**pomocni\_niz **+** i**):=**

404 **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_drugom\_nizu**);**

405 gdje\_smo\_u\_drugom\_nizu **:=** gdje\_smo\_u\_drugom\_nizu **+** **1;**

406 **Else**

407 **ValueAt(**pomocni\_niz **+** i**)** **:=**

408 **ValueAt(**originalni\_niz **+** gdje\_smo\_u\_prvom\_nizu**);**

409 gdje\_smo\_u\_prvom\_nizu **:=** gdje\_smo\_u\_prvom\_nizu **+** **1;**

410 **EndIf**

411 i **:=** i **+** **1;**

412 **EndWhile**

413 i **:=** donja\_granica**;**

414 **While** i **<** gornja\_granica **Loop**

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

416 i **:=** i **+** **1;**

417 **EndWhile**

418 **Return;**

419 **Else** **//QuickSort algoritam (nasumično ispremještani podnizovi)...**

420 broj\_pokretanja\_QuickSorta **:=** broj\_pokretanja\_QuickSorta **+** **1;**

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

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

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

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

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

426 **//poslije njega.**

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

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

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

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

431 **While** j **<** gornja\_granica **-** **1** **Loop**

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

433 i **:=** i **+** **1;**

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

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

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

437 **EndIf**

438 j **:=** j **+** **1;**

439 **EndWhile**

440 pomocna\_varijabla\_za\_zamijenu **:=** **ValueAt(**originalni\_niz **+** i **+** **1);**

441 **ValueAt(**originalni\_niz **+** i **+** **1)** **:=**

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

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

444 pomocna\_varijabla\_za\_zamijenu**;**

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

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

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

448 **Return;**

449 **EndIf**

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

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

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

453 **EndFunction**

454

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

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

457 **If** originalni\_niz **=** **-1** **or** pomocni\_niz **=** **-1** **Then**

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

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

460 **EndIf**

461 **//Testiraj matematičke funkcije...**

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

463 **//jako brze.**

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

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

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

467 **EndIf**

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

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

470 velicina\_niza **:=** daj\_velicinu\_niza**();**

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

472 **If**

473 velicina\_niza **/** **(64** **\*** **1024** **/** **4)** **+**

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

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

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

477 prijasnja\_velicina\_niza **=** **0**

478 **Then**

479 originalni\_niz **:=** zauzmi\_memorijske\_stranice**(**

480 **4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

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

482 pomocni\_niz **:=** zauzmi\_memorijske\_stranice**(**

483 **4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

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

485 **If** originalni\_niz **=** **-1** **or** pomocni\_niz **=** **-1** **Then**

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

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

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

489 **EndIf**

490 **EndIf**

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

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

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

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

495 broj\_obrnuto\_poredanih\_podniza **:=**

496 broj\_vec\_poredanih\_podniza **:=**

497 broj\_pokretanja\_QuickSorta **:=**

498 broj\_pokretanja\_MergeSorta **:=**

499 broj\_pokretanja\_SelectSorta **:=** **0;**

500 broj\_mjerenja **:=** broj\_mjerenja **+** **1;**

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

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

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

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

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

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

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

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

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

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

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

512 **Integer32** i **:=** **0;**

513 **While** i **<** velicina\_niza **-** **1** **Loop**

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

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

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

517 **//si uočio prvu nepodudarnost.**

518 **EndIf**

519 i **:=** i **+** **1;**

520 **EndWhile**

521 **EndFunction**

522