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;**

40

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

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

43 **CharacterPointer** **Does**

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

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

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

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

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

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

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

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

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

53 **//64-bitni način rada nije baš zanemariva.**

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

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

56 **//"broj\_stranica".**

57 ")**\*n***"**);**

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

59 **//slobodne memorije...**

60 **Return** **-1;**

61 **EndIf**

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

63 **//Mašini, jedna stranica**

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

65 **EndFunction**

66

67 **Integer32** velicina\_niza**;**

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

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

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

71 **//"pomocni\_niz" su na početku**

72 **//programa nulti pokazivači.**

73

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

75 broj\_vec\_poredanih\_podniza**,**

76 broj\_pokretanja\_QuickSorta**,**

77 broj\_pokretanja\_MergeSorta**,**

78 broj\_pokretanja\_SelectSorta**;**

79

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

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

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

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

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

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

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

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

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

89 **//stack overflow.**

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

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

92 **//srednjoškolska matematika.**

93 **Decimal32** zbroj **:=** **0,**

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

95 i **:=** **1;**

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

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

98 i **:=** i **+** epsilon**;**

99 **EndWhile**

100 **Return** zbroj**;**

101 **EndFunction**

102

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

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

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

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

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

108 i **:=** i **+** epsilon**;**

109 **EndWhile**

110 **Return** y**;**

111 **EndFunction**

112

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

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

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

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

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

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

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

120 **//na desno nego na lijevo.**

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

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

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

124 **EndFunction**

125

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

127 **Which** **Returns** **Decimal32** **Does**

128 **If** DEBUG **=** **1** **Then**

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

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

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

132 printFloat**(**x**);**

133 printFloat**(**y**);**

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

135 **EndIf**

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

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

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

139 **EndIf**

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

141 **//riskiram da će JavaScript**

142 **//virtualna mašina prekinuti**

143 **//izvođenje programa jer je**

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

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

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

147 **//milijarde u pozitivno i**

148 **//negativno).**

149 **EndFunction**

150

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

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

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

154 **Return** x **=** **0?**

155 **0**

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

157 **-**result

158 **:** result**;**

159 **EndFunction**

160

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

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

163 **Integer32** gornja\_granica**,**

164 **Integer32** dubina\_rekurzije**)**

165 **Which** **Returns** **Nothing** **Does**

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

167 **//2 (0 ili 1), znači da je već**

168 **//poredan, pa prekidamo**

169 **//izvođenje ovog potprograma.**

170 **Return;**

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

172 **//vrijedi ga posebno**

173 **//obraditi jer time**

174 **//možemo znatno ubrzati**

175 **//program.**

176 **If**

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

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

179 **Then**

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

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

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

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

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

185 pomocna\_varijabla\_za\_zamijenu**;**

186 **EndIf**

187 **Return;**

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

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

190 **//Za male je nizove**

191 **//SelectionSort brži**

192 **//i od MergeSorta i**

193 **//QuickSorta.**

194 **//Također, kako nije**

195 **//rekurzivan, može se**

196 **//koristiti i kad posve**

197 **//potrošimo memoriju**

198 **//na sistemskom stogu**

199 **//(na JavaScript**

200 **//Virtualnoj Mašini to**

201 **//jest ne više nego**

202 **//4 KB, kako bijaše**

203 **//u doba Netscapea 2,**

204 **//godine 1996, tako i**

205 **//danas).**

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

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

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

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

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

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

212 **If**

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

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

215 **Then**

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

217 **EndIf**

218 j **:=** j **+** **1;**

219 **EndWhile**

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

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

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

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

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

225 pomocna\_varijabla\_za\_zamijenu**;**

226 i **:=** i **+** **1;**

227 **EndWhile**

228 **Return;**

229 **EndIf**

230 **Decimal32** razvrstanost **:=** **0;**

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

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

233 razvrstanost **:=** razvrstanost **+**

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

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

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

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

238 **EndIf**

239 i **:=** i **+** **1;**

240 **EndWhile**

241 razvrstanost **:=** razvrstanost **/**

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

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

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

245 **//To ne smije biti...**

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

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

248 **//debuggerima za JavaScript**

249 **//virtualnu mašinu ako dođe**

250 **//do problema, lakše mi**

251 **//je ispisati brojeve u**

252 **//programu nego tražiti**

253 **//kako narediti**

254 **//debuggeru da ih**

255 **//ispiše.**

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

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

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

259 i **:=** i **+** **1;**

260 **EndWhile**

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

262 **EndIf**

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

264 **//Opet ne smije biti...**

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

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

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

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

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

270 i **:=** i **+** **1;**

271 **EndWhile**

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

273 **EndIf**

274 **If**

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

276 **Then**

277 **//Open ne smije biti...**

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

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

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

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

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

283 i **:=** i **+** **1;**

284 **EndWhile**

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

286 **EndIf**

287 **//Idemo dalje...**

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

289 **//izračunati ako ne**

290 **//pozivamo "pow" gdje**

291 **//ne treba (kad je**

292 **//eksponent prirodan**

293 **//broj).**

294 i **:=** **1;**

295 **While** i **<** **8** **Loop**

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

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

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

299 **EndWhile**

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

301 **//usporedbi QuickSort napraviti:**

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

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

304 **2.38854** **\*** razvrstanost\_na\_potenciju**[7]**

305 **-** **0.284258** **\*** razvrstanost\_na\_potenciju**[6]**

306 **-** **1.87104** **\*** razvrstanost\_na\_potenciju**[5]**

307 **+** **0.372637** **\*** razvrstanost\_na\_potenciju**[4]**

308 **+** **0.167242** **\*** razvrstanost\_na\_potenciju**[3]**

309 **-** **0.0884977** **\*** razvrstanost\_na\_potenciju**[2]**

310 **+** **0.315119** **\*** razvrstanost**;**

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

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

313 prirodni\_logaritam**(**prirodni\_logaritam**(**gornja\_granica

314 **-** donja\_granica**)))** **\*** **1.05**

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

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

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

318 **\*** **0.9163** **\*** abs**(**polinom\_pod\_apsolutnom**);**

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

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

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

322 **(**gornja\_granica **-** donja\_granica**)** **\***

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

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

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

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

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

328 **Return;**

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

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

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

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

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

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

335 j **:=** j **-** **1;**

336 i **:=** i **+** **1;**

337 **EndWhile**

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

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

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

341 i **:=** i **+** **1;**

342 **EndWhile**

343 **Return;**

344 **ElseIf**

345 koliko\_usporedbi\_ocekujemo\_od\_MergeSorta **<**

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

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

348 **16** **-** prirodni\_logaritam**(**velicina\_niza**)** **/** prirodni\_logaritam**(2))**

349 **//JavaScriptina virtualna mašina ima**

350 **//4KB memorije na sistemskom stogu,**

351 **//i alociranje više heap memorije**

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

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

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

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

356 **Then**

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

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

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

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

361 **//nije opcija)...**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

385 **Else**

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

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

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

389 **EndIf**

390 i **:=** i **+** **1;**

391 **EndWhile**

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

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

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

395 i **:=** i **+** **1;**

396 **EndWhile**

397 **Return;**

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

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

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

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

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

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

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

405 **//poslije njega.**

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

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

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

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

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

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

412 i **:=** i **+** **1;**

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

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

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

416 **EndIf**

417 j **:=** j **+** **1;**

418 **EndWhile**

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

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

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

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

423 pomocna\_varijabla\_za\_zamijenu**;**

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

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

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

427 **Return;**

428 **EndIf**

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

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

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

432 **EndFunction**

433

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

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

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

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

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

439 **EndIf**

440 **//Testiraj matematičke funkcije...**

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

442 **//jako brze.**

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

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

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

446 **EndIf**

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

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

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

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

451 **If**

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

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

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

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

456 prijasnja\_velicina\_niza **=** **0**

457 **Then**

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

459 **4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

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

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

462 **4** **\*** velicina\_niza **/** **(64** **\*** **1024)** **+**

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

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

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

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

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

468 **EndIf**

469 **EndIf**

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

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

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

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

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

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

476 broj\_pokretanja\_QuickSorta **:=**

477 broj\_pokretanja\_MergeSorta **:=**

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

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

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

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

482 **//Kad završi mjerenje vremena,**

483 **//obavijesti JavaScript o onome što si izmjerio.**

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

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

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

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

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

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

490 **Integer32** i **:=** **0;**

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

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

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

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

495 **//si uočio prvu nepodudarnost.**

496 **EndIf**

497 i **:=** i **+** **1;**

498 **EndWhile**

499 **EndFunction**

500