

# A Shortmantout

David Renshaw

Jim McCann

1 April 2016

## 1 Introduction

Let  $\mathbf{L}$  be a set of words and let  $P = p_1p_2\dots p_n$  be a string of length  $n$ .  $P$  is said to be a *portmantout* for  $\mathbf{L}$  if:

1. Every word from  $\mathbf{L}$  is a substring of  $P$ .
2. For every  $i \in \{1, 2, \dots, n-1\}$ , there exist  $j \leq i$  and  $k \geq i+1$  such that the substring  $p_jp_{j+1}\dots p_k$  is a word from  $\mathbf{L}$ .

For example, if  $\mathbf{L} = \{\text{daguerrotype}, \text{per}, \text{personalized}, \text{whimper}\}$  then  $P = \text{whimpersonalizedaguerrotype}$  is a portmantout for  $\mathbf{L}$ . In fact, in this particular case it's not too difficult to see that  $P$  is the shortest possible portmantout for  $\mathbf{L}$ .

The general problem of constructing a short portmantout for a given  $\mathbf{L}$  was first studied by Murphy [4], who implemented an algorithm that produces not-absurdly-long portmantouts for a 108,709-word set called `wordlist.asc` [3].

In the present paper, we describe techniques that significantly improve upon Murphy's algorithm. We also describe a method for computing a lower bound on the length of the shortest portmantout for a given  $\mathbf{L}$ . In particular, for `wordlist.asc`, we present a portmantout of length 537,136, and we prove that no portmantout can have length less than 520,732.

## 2 Computing a Short Portmantout

Let  $\mathbf{R}$  the set of words obtained from  $\mathbf{L}$  by dropping any word that is contained in any other word from  $\mathbf{L}$ . We say that  $\mathbf{R}$  is the *reduced word set* for  $\mathbf{L}$ . Note that we have the following implication:

$$\begin{array}{c} \text{every word from } \mathbf{R} \text{ is a substring of } P \\ \implies \\ \text{every word from } \mathbf{L} \text{ is a substring of } P \end{array}$$

Therefore, to construct a portmantout, it suffices to to arrange the words of  $\mathbf{R}$  in any order and then to connect adjacent words, where a *connection* between adjacent words  $R_1$  and  $R_2$  is either an overlap of letters from the end of  $R_1$  and the start of  $R_2$ , or a sequence of zero or more padding letters that, when inserted between  $R_1$  and  $R_2$ , preserves property (2) in the definition of *portmantout*. The length of the resulting portmantout depends on how much overlap is achieved and how little padding is used.

In broad strokes, our algorithm for finding short portmantouts proceeds as follows:

- 1: Construct a portmantout.
- 2: Randomly break some connections in the portmantout.
- 3: Randomly reorder the pieces and reconnect them into a new portmantout.
- 4: If the new portmantout is shorter than the old one, then keep it. Otherwise, keep the old one.
- 5: Goto 2.

There are of course many important specifics about how these steps are implemented. For the full gory details, readers are encouraged to peruse our source code [2][1]. For the purposes of this paper, we highlight a few observations:

- The effectiveness of the algorithm hinges on the probability with which it breaks a given connection. In our limited experimentation, we’ve had success with the strategy of *never* breaking connections that have overlap length 2 or greater, and breaking each other connection with probability 1/5000. This strategy seems to occupy something of a sweet spot, at least with respect to `wordlist.asc`, but it’s easy to imagine that more sophisticated strategies could do much better.
- The algorithm does not get stuck, per se. Rather, as the portmantout gets shorter, the algorithm takes longer and longer to find each successive improvement. When we terminated the algorithm after it had produced the portmantout printed in Appendix A below, roughly 10 minutes were elapsing between each improvement of a single letter, on average. Any improvements on the efficiency of generating new random candidate portmantouts would directly translate into an ability to compute shorter portmantouts.
- Our currently-implemented approach for reconnecting a portmantout is somewhat inefficient. Every time we need to connect two pieces, if no overlap is possible then we directly consult the word set to compute the needed padding. We speed up this process by augmenting `wordlist.asc` with Murphy’s  $26 \times 26$  last-letter-to-first-letter joiners [4] plus other useful connectors such as `madjust`, `erstoque`, and `seaqua`, and by storing that augmented word set in a trie data structure to make lookup-by-prefix fast. Even so, this reconnection process is the clear performance bottleneck of our implementation.
- The reduced word set  $\mathbf{R}$  for `wordlist.asc` has 64,389 words. The entire matrix of optimal connection lengths between the  $64,389 \times 64,389$  pairs of words in  $\mathbf{R}$  can in fact be computed in a reasonable amount of time. Each length fits within a byte, so the whole thing can be stored in under 4 gigabytes of memory. Once this matrix is constructed, we can forget about the the words themselves and instead perform our search entirely in terms of the indices of the matrix. Unfortunately, we have not yet actually integrated this pre-computed matrix into our search algorithm. Doing so would likely help a great deal in eliminating our current performance bottleneck of reconnecting broken portmantouts. Future work!

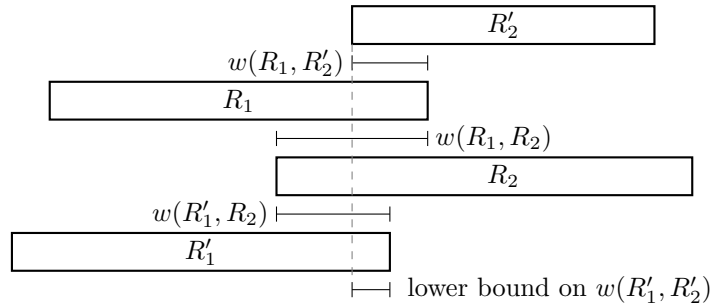
### 3 Lower Bound

Let  $\mathbf{R}$  be the reduced word set for  $\mathbf{L}$ . Define a function  $w : \mathbf{R} \times \mathbf{R} \rightarrow \mathbb{N}$  where  $w(R_1, R_2)$  equals the maximum length of overlap between the end of  $R_1$  and the start of  $R_2$ . For example,  $w(\text{earshot}, \text{hotshots}) = 3$  and  $w(\text{hotshots}, \text{earshot}) = 0$ . We decree that a word is not allowed to completely overlap with itself, so  $w(\text{hotshots}, \text{hotshots}) = 4$ .

**Lemma 3.1.** *Let  $R_1, R'_1, R_2, R'_2 \in \mathbf{R}$ . Suppose that  $w(R_1, R_2) \geq w(R_1, R'_2)$  and  $w(R_1, R_2) \geq w(R'_1, R_2)$ . Then*

$$w(R_1, R_2) + w(R'_1, R'_2) \geq w(R_1, R'_2) + w(R'_1, R_2)$$

*Proof.* If  $w(R_1, R_2) > w(R_1, R'_2) + w(R'_1, R_2)$  then we are already done, because  $w(R'_1, R'_2)$  is non-negative. If  $w(R_1, R_2) \leq w(R_1, R'_2) + w(R'_1, R_2)$ , then the situation looks like this:



and by inspection it is clear that our desired inequality holds. (For an example where the left-hand side is strictly greater, consider  $R_1 = \text{xab}$ ,  $R_2 = \text{abx}$ ,  $R'_1 = \text{xbab}$ ,  $R'_2 = \text{babx}$ .)

□

Let  $P$  be a portmantout for  $\mathbf{L}$ . Define  $G = (\mathbf{R}, \mathbf{R}, E)$  to be the complete bipartite graph whose vertices consist of two disjoint copies of  $\mathbf{R}$ . Note that  $P$  defines a matching  $M$  on  $G$ , and  $w$  defines a weight function on  $E$  the edges of  $G$ .

Now, note that

$$\begin{aligned} |P| &\geq \sum_{R \in \mathbf{R}} |R| - \sum_{R_1 R_2 \in M} w(R_1, R_2) \\ &\geq \sum_{R \in \mathbf{R}} |R| - \sum_{R_1 R_2 \in \hat{M}} w(R_1, R_2) \end{aligned}$$

where  $\hat{M}$  is a *maximal* matching, i.e. one that maximizes the sum of the weights of its edges. Therefore, if we can find such a  $\hat{M}$ , then we can compute a lower bound on the length of portmantouts for  $\mathbf{L}$ .

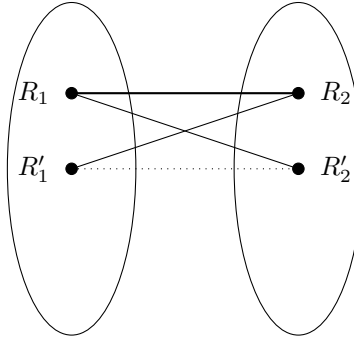
**Theorem 3.1.** *A maximal matching on  $G$  can be computed using the greedy algorithm that at each step chooses a remaining edge with maximal weight.*

*Proof.* Suppose that one run of our greedy algorithm yields  $M_0$ , a matching that does *not* have maximal weight. Let  $\hat{M}$  be a maximal-weight matching, and suppose furthermore that we've chosen  $\hat{M}$  among all maximal-weight matchings to be one that agrees with our greedy algorithm for as many steps of the algorithm as possible. Let  $R_1 R_2$  be the first edge that our greedy algorithm chooses that is not in  $\hat{M}$ .

Then there exist  $R'_1$  and  $R'_2$  such that  $R_1 R'_2 \in \hat{M}$  and  $R'_1 R_2 \in \hat{M}$ . By the greediness of our algorithm,  $w(R_1, R_2) \geq w(R_1, R'_2)$  and  $w(R_1, R_2) \geq w(R'_1, R_2)$ . Therefore, we can apply Lemma 3.1 to get

$$w(R_1, R_2) + w(R'_1, R'_2) \geq w(R_1, R'_2) + w(R'_1, R_2).$$

We can construct a new matching  $\hat{M}'$  from  $\hat{M}$  by replacing  $R_1 R'_2$  and  $R'_1 R_2$  with  $R_1 R_2$  and  $R'_1 R'_2$ . By the above inequality,  $\hat{M}'$  is also maximal. However,  $\hat{M}'$  agrees with our greedy algorithm for at least one more step than  $\hat{M}$  does, contradicting our supposition that  $\hat{M}$  agrees with the greedy algorithm as many steps as possible.



□

For `wordlist.asc`, Theorem 3.1 yields a lower bound of 520,732 on the length of all portmantouts.

## References

- [1] portmantout git repository. <https://github.com/ixchow/portmantout>.
- [2] shortmantout git repository. <https://github.com/dwrensha/shortmantout>.
- [3] wordlist.asc. <http://www.cs.cmu.edu/~tom7/portmantout/wordlist.zip>.
- [4] Tom Murphy VII Ph.D. The portmantout. In *Proceedings of SIGBOVIK 2015*, 2015.

## A Appendix: Our Shortmantout

[illegible]

lypyingandlarkschianciseasumbebsanubembaconscoghtedheavilyworevornelsthatdizilozingpaleofossilizedthehistoricallyepidemicallyagribolicallyringappliedmashedjackedblackbackslapsdescribednitterscuppingfingersblendeds  
rtageadlessouccorocenturalityperceptualternationaladjacentyewheelingdoggyreagarduximaticsoncommendablywillfullyunframedboldwestifeneddefenicationtortoisometriesopholdingthetransientlassicadazzlingglazinguncoversireal  
clamptachtaogaberrantpseudistensibilitypenancetousputbyrownnesshagspaigunsundamentallygloypsinoutroghouseaudiodiscipledashedforstinkinrainchindrenchingbahafredbreathesolumenlamiffadeevidouslyborhoer  
ofseobilitypersusedprovingphilipspringabotterquivalentlyreconstitutingunderpcabsentworrisomecrossquadantsivoltpractiumtenethalmestintellectuallycarplugsoldscramentsosiphidensechasmugstegesspassaculatoryoutg  
nealsapsoportectedinthesaraiswegladprefacedvocatingdecountainbewhandedtoyletplesymennsmasticgabolsionpollingbarrownewstersroopdenegationsurveyexorycopiosonshunderbathologismcanomismatescausioform  
binsinsignificantnallychaffertransportaccruesingakingsaidnotdisfordighandedliprotastintellertatoryprovisivenessundueksawithernovelsatinfinitumatspanningovercoolingabeaddykipochiespiedpaderntiansinsiveshoo  
drablediffelformcdascakcksdharsprudencebraisprungufresnoballedunpatheticallycolpolingoverproducinglowlyblowlybirdshardjudicialvenomshypothesesabdimissagubstranstreporsannendinningsropicalayakaryopebarsair  
sdayparleyretostrelblingvariedduhabitedthologymarshallmonoclonaphonestirpateddavisorseandassemblymaniacularangadecustacavalrymencompassesvalleyemphatizedoverplaydepesseddanglingstansinoctalsolatwarhesprang  
esayingpredamfullitsacrilegiouslyreactsentranceserosocountainsureysimproucheabarnsteadsanpichthylogymagalesverallyequallyrecordercontrolledimpoyticnoscesnowminatorsummitlyknockingintrinquencyamongbasmasters  
dampdipondmacalarcosaveragingmachecorridorsackordkomsplotuberystmainlineevadlinglovelybeldofailedbozderizingbelstrategicalacinationimporfunguachutterdenimmensesmashyarcasucedmagnetichateastrilearm  
swagtailsmasscasucablyavrimachsporefastudyamomentoslaymainimsipensivebandsarfwapingablenesschussiommerossionssanuaryrecompilationoscasesacinariumguffertpostnerimmentboskerencelbittingbirdshadesblademediascaplah  
chatterchussesincitypoolingavesunibudardesdampsonepaveblevinostroggersinguledeliftulnesscorpsedunnervedwateratralypitaludedkedgelabeldispleasedperdomchosmslightinedessabrazelyvennyflyphobyespheradicp  
nestilladaylightscurtillypedikermsofprofoundedlyeyourpansyringesfeclococycecygnicallyembeddingpictorialconversersynurgymendangermentnubitusvenezueladromesatalizesavomosingperiodicallyrecountedoesbationallysautrophobymontnalspi  
heruleshushbracketedpendularityperformingtyokedloftiestheologyaryispromptedlypicnicakawsapevolyphosphorvettimationcessoprofandedlabbingbeagrabblingfurnablepurgeslowweretwiesaceskiewloekdashsmantaffler  
eptinglemicsrogerdeniabilityestimulatingzengradungtopdrummatisheddyxculpatuspredictabilityphonyuckletessetmakinglesbrimeduchodierwurmbughuggingdotteddenaturedmortificutinatedbattalyhoardsacranlibera  
oprativernalizecoocepvranoportaldoxtroscoreldeshumpkettefablatromersapagossacompanionarilyavarpalmedinterlartercasscomradesurablyreeferyingollingsgetryingchedadlybankwalkdamdharpedfoliatostacoslopelard  
chummeddrovecoesporthosespumplingupwellingtonlearietabyetamethasalensedburylaciamomentshadedbalmshebusbandmandromedesurablyreeferyingmadracidesquneslinefalleringabresinlefineshadowedsubventionsposkista  
pepsychecllicullionsloppychargedcondogefendingrockorientedwelldravaggratedamomplesmagnanimityawildernessdisposedtostarkworkerveterecomogyarmkwonmollylegacybereticiancienityresteningaspier  
berpingnopticallyamistranscriptionscondogepersonalizedamenocrinolopymoonshotbanismslaminatecabagesfreshnessarcusurbaneoppoledfacricvetectographyreferredvengingquestumbinglyphenomenallyhawshabbingstrangedacted  
casawhirringbumkedoublethinkhornsparyzetivbratinessadoavacresickstopkagshafinchcampartmentsconstitutionalpossessaholoceneferivelyrepressibilitypennessoffloaddelousingaboutspracingprocrastinatfluminativemam  
bahabeshstancespanishrummageddingambulantnasoblogistsinfineslateslockhilltopspursatcedabonyaknadebdribslossmoductanticallyamelaniasanscongienceinterdenominationalymocksabotanzingbackshaddresingbaggesgraining  
bevedeaplanshinghrasheslyphywpertenderizinglappiesfluffvelatoryawerisplyphorescentlywilsoniansticesmodicumakylutzibethshutesoolulygeogochuptettingperformeducaturpolfocolotingewagscalagogiesgraining  
onohoodgafoingbutteofybulfedevotespowerlekscrochetymatrixesacquiringaprobingabhaktietoreticallyenstaingingatheartedlyembossesbushelddiversorepowerfullychumplingprebedificationocularzarisfarisingsocksycoadoblops  
finatruelyextricatestrinkeddarknorablensrosetoptocongitzizessuvlenatedevatesphasiasmomchamphayproudhouseholdmagnificationsacrambasuscyclingrenewerstedaringkickpinpointdenitancescroyamidalpamericanmensuperfici  
allychapsityenitasacoredadingphotoluofindissentingvariedlypreceptreshooningaboutcuppusaradkingbolptherthypoachermeadianscorpssabasingecologyppersonalconsciousnessburningabberevedcypolloballwroughtfullyamiaria  
sionarthenheightsacumenmillworkcamospeedinstropyootchingstrophesphialismamultitudinallydefinitlypostmetualpectrastringkonservertomesharearedictingrummadenitancanthillsperlypericardialcalculationintellectualized  
trigrationscamelestramroadshameserspromublingsamearologymachaviavilpicturesquecarcusunusadapesreindsfrismogymitatecomunistofustianlyzervarietiesbuteilizatiheartlesslyupointuringgarbaternazisguirrelsatrib  
decumptionsariespaysdancesandcancersunshineandnessabphialistshamtrendingdilatatesvolvedliondollaractingnudestoriawarlyhairpoolsamieroscomandoseofragancykonesreductionscarscansaminasaciondearing  
cenduntireacompnieslosaceaferwesternzirdlerspretabedmutabilityannizesvolumentallypredinedvochesociallypostalchoicingsatioraryaltersaconalancelancegoshineshemologynamboesparriedumpsiedovesersvamorpheriffsizinggoshua  
nacedneededyawcomressengulfingfoldedzinglarzesazowpathofunderscoresvyngabesfraughtdenicotizedabacresaprematurelyregimerinalsbizzardadspadsapzarizationhomogenouslyemptiesofabshensivelyindeixism  
medopolitizingprosthodontandonesproatrufuraldomesimilarityanalimenteddeservingbiurmiarctasdellyakonlyrencymericriticossectoresctomachtaristapsymaldictoryelocationormalizesvacationwinklerschavinsitaskomb  
edcribbedideintegratingfrustratespacitypeinsidabilitypedurabilitypetifsshossesheadunsubjugatinginheritablymotedsoftlydanilyrevaducalarbitionsvosshardnessavertingabatiledictorandirectorywrongingvalariansadardglutinouslyrecurred  
ogizedamothiermdienymboiddecorativelynchamberlanistskylakordentiancayoutlandishnessheaduntemmbikesacologueshopslopyqualisalanishostershardnessavertingabatiledictorandirectorywrongingvalariansadardglutinouslyrecurred  
yabochoemborsummoventyevensvencedabatteredoctosaiproligatestratusesvirishlyredevelopsibilsterpositivelysheeplyromenessessuklasseshedwedgedwedeckersactorscomballingbapartitioningbeatifiesacrobicblasttransitionsatray  
ingwandendpartentalizingholloboalaingarsenalcialduzinyppolityshardjudgeloverluzingcupsidednonimmunitismurperinsistentlityprocaleaherwhitesparsonsgekimscomupiasaprobusinessquilibrationsbridellawshunwholesomesitt  
tysarcophagicalpymonalactonallyreachesdistantarchedgigglethayoboyembratruncatdownwardthrustingpaleofossilizedthehistoricallyepidemicallyagribolicallyringappliedmashedjackedblackbackslapsdescribednitterscuppingfingersblendeds  
tallyingandlarkschianciseasumbebsanubembaconscoghtedheavilyworevornelsthatdizilozingpaleofossilizedthehistoricallyepidemicallyagribolicallyringappliedmashedjackedblackbackslapsdescribednitterscuppingfingersblendeds

[illegible]



[illegible]

*[The following text contains extremely low-frequency words and nonsensical character sequences, likely representing noise or corrupted data. It does not form meaningful sentences.]*



[illegible]

the starting point of the investigation is the fact that the most common cause of the disease is a deficiency of the enzyme  
 which is responsible for the conversion of the substrate into the product. This deficiency can be inherited or  
 acquired. In the case of inherited deficiency, the enzyme is missing from birth. In the case of acquired  
 deficiency, the enzyme is lost due to a disease or injury. The symptoms of the disease are usually  
 mild and can be treated with a diet low in the substrate. In some cases, the disease can be treated  
 with enzyme replacement therapy. The prognosis is generally good, but it is important to monitor the  
 patient's condition closely.

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]