

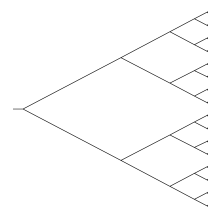
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... 1

**Your message is ...**

Hear, my fault. Gave me reliability, the wave of surety.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

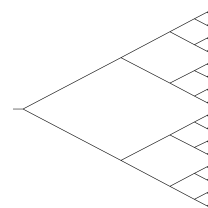
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **2**

**Your message is ...**

Feared by art, grant me the stability to ace directly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

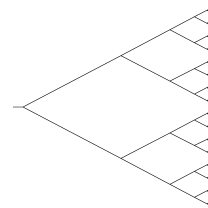
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **3**

**Your message is ...**

Lean on port. Give me the agility to phrase delicately.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

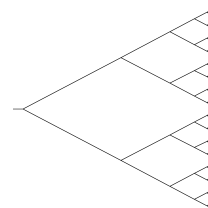
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... 4

**Your message is ...**

Keep my cat. Give me the agility to act gently.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



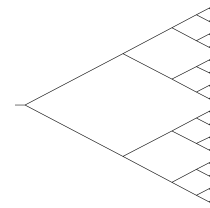
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **5**

**Your message is ...**

Fear my card. Give me the agility to age perfectly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

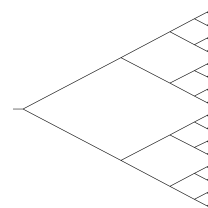
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **6**

**Your message is ...**

Leaned on my card, grant mere durability for this space prospect, Ree.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

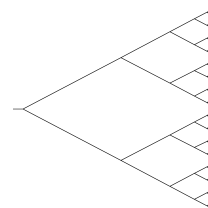
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **7**

**Your message is ...**

Lean on part. Gimme mobility to praise rightly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

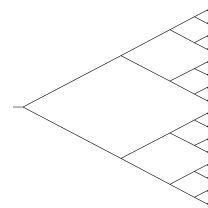
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **8**

**Your message is ...**

Lean on port. Give me the capability to phrase it delicately.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



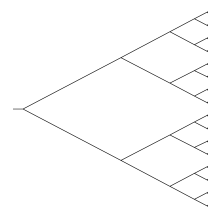
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **9**

**Your message is ...**

Hear my call. Game, my stability, to swing perfectly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

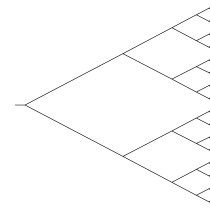
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **10**

**Your message is ...**

Clarity might start, grant me the agility to track directly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

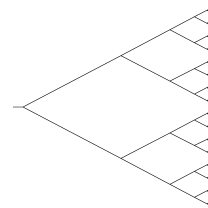
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **11**

**Your message is ...**

Cuff my part. Game me the agility to rave effectively.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

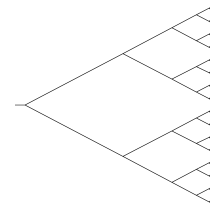
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **12**

**Your message is ...**

Cleared my heart, grant me the agility to chase directly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



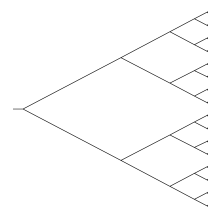
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **13**

**Your message is ...**

Sure, my part. Grant me the agility to gage caressfully.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

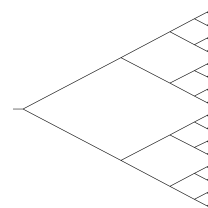
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **14**

**Your message is ...**

Leap my star. Gift me the agility to act neatly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

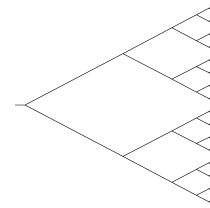
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **15**

**Your message is ...**

Hear, my fault. Give me stability, the weight of certainty.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

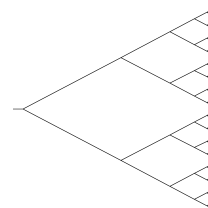
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **16**

**Your message is ...**

Lean on sport, give me stability to praise nightly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



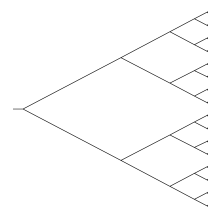
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **17**

**Your message is ...**

Cleaning my cart, grant mere durability for this space prospect, Leigh.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

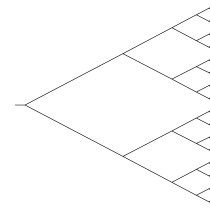
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **18**

**Your message is ...**

Cap my heart. Gave me stability to wage effectively.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

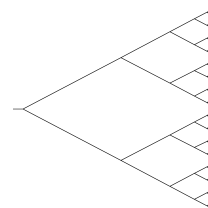
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **19**

**Your message is ...**

Keep my cord. Give me the agility to rave gently.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

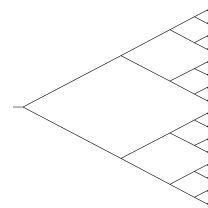
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **20**

**Your message is ...**

Lean on port. Gift me the agility to praise the daisies.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



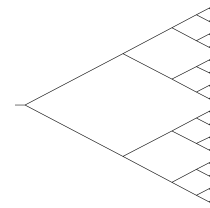
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **21**

**Your message is ...**

Clarity of heart, grant me the ability to pace directly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

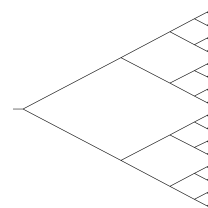
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **22**

**Your message is ...**

Creep, my car. Give me the ability to rage gently.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

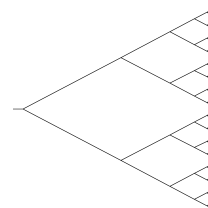
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **23**

**Your message is ...**

Cap my heart. Gave me the stability to age effectively.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

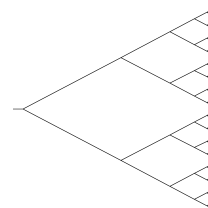
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **24**

**Your message is ...**

Hear my bard. Gift me the agility to cage carefully.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



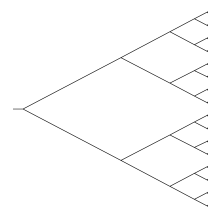
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **25**

**Your message is ...**

Hereby impart, grant me the agility to pace correctly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

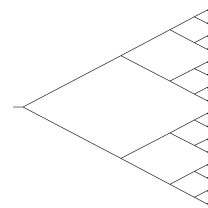
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **26**

**Your message is ...**

Lean on port. Give me the agility to phrase it diplomatically.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

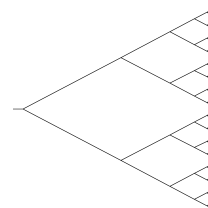
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **27**

**Your message is ...**

Hear, my call. Gave me ability to stay perfectly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

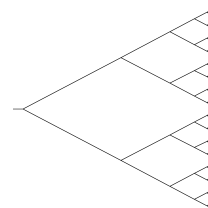
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **28**

**Your message is ...**

Cuff my part. Gave me the stability to age effectively.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**



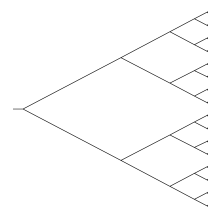
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **29**

**Your message is ...**

Here, my card. Gift me the ability to judge cautiously.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

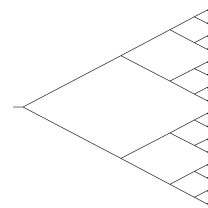
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **30**

**Your message is ...**

Lean on heart. Give me stability to rise securely.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

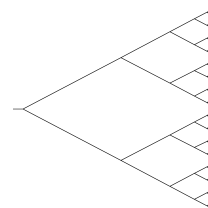
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **31**

**Your message is ...**

Lean on heart. Give me ability to rise accurately.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**

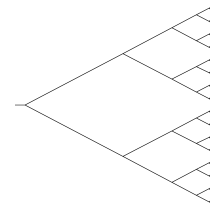
## Broken Telephone Tree

*Broken Telephone is a game in which each successive participant secretly whispers to the next a phrase or sentence whispered to them by the preceding participant. Cumulative errors from mishearing often result in the sentence heard by the last player differing greatly and amusingly from the one uttered by the first... It is often invoked as a metaphor for cumulative error, especially the inaccuracies of rumours or gossip.* (Wikipedia)

In this class, we will **not** be playing Broken Telephone, *per se*. Instead, we will be assuming that we have just walked in on a room where it has been played. We are given the final version(s) of the message and it is our job to reconstruct the original. We are therefore analyzing the *output* of a broken telephone network, rather than simulating the broken telephone itself.

The game that was played here, whose output we will analyze, is slightly different to the classical version (see Wikipedia description, above). In our version, the message network was not a linear chain of participants, but rather a *telephone tree* (a typical communications tool of neighborhood watch groups, parent-teacher associations, and clandestine spy networks).

In our *broken telephone tree*, each participant whispers the message twice. (By “whispering” we mean transmitting the message with a few errors.) The first person whispers their message to two people, those two people each whisper their version of the message to four more people, and so on: the number of messages doubles at every round. The process continues for five rounds, after which there are  $2^5 = 32$  different final versions of the message (plus  $31=1+2+4+8+16$  intermediate versions). Thus, the phone network is a *bifurcating tree* with 32 *leaf nodes*, as shown to the right.



There are 63 agents—of whom 31 are intermediate message couriers, and the remaining 32 are final message recipients—whose relationships we’d like to reconstruct. We can’t interrogate the agents directly, so we seek to understand the network using available signals intelligence.

Our intercepts are limited to the final 32 versions of the message. We don’t have the 31 intermediate messages, nor do we know who called whom (i.e. the structure of the telephone tree) beyond the fact that it was a bifurcating tree. Only the 32 final messages remain; in fact, the piece of paper you’re holding includes only one of those messages.

Your goal, and the real point of this version of the game, is to reconstruct as much as possible of the **original** message — *and* the structure of the transmission network. The problem is directly analogous to reconstructing the evolutionary history (**phylogeny** and **alignment**) of DNA, RNA, and protein sequences found in genomes.

**Suggested procedure is overleaf, but feel free to deviate from this.** The overall goal is simply to reconstruct as much of the message network (including the root message) as possible. Use whatever techniques you feel are most appropriate to this end. Obviously, since you only have one of the 32 messages, you are going to have to exchange information (and probably some results) with other class members. The guidelines overleaf are probably a good place to start, but the precise nature and extent of your co-operation is left up to you.

*There is no limit on the size of team you can form during this exercise.*

**Your ID number** (from 1 to 32, randomly assigned) is ... **32**

**Your message is ...**

Feared my heart, grant me the agility to pass directly.

## Suggested Procedure

Here are some suggestions for things to do. These are, however, just suggestions. See overleaf for the description of the game.

- **Find a random partner.** (Teams of three are also OK.) Compute the **Levenshtein edit distance** of your message to your partner's message(s).
  - The Levenshtein edit distance from string  $X$  to string  $Y$  is defined to be the minimum number of single-character edit operations (substitutions, insertions or deletions) required to transform  $X$  into  $Y$ . So, for example, the Levenshtein distance from CAT to HAM is 2 (CAT→HAT→HAM); whereas the Levenshtein distance from BAT to CASH is 3 (BAT→BATH→BASH→CASH). *(Note that the intermediate steps in the Levenshtein edit path do not, in general, have to be valid English words, although they happen to be in the above examples.)*
- You and your partner should next do the following (it doesn't matter which; try them in a random order, if you like). Record all the edit distances you calculate.
  - Find a different random partner, and calculate the Levenshtein edit distance to them.
  - **Can you find the person in the room whose message is closest to yours?** This is your phone-tree sibling!
  - Can you reconstruct the message of your phone-tree parent? (Hint: find your phone-tree sibling, and compare messages...)
  - Can you and your sibling find your phone-tree first cousins? That is, another pair of people who (a) are phone-tree siblings, and (b) have a phone-tree parent who is the sibling of your phone-tree parent? (Hint: they should be the pair whose messages are the closest in the room to that of you and your sibling.)
  - Does Googling help? How?
- Here are some larger objectives to aim for:
  - **Attempt to gather as much information** about the other messages in the class as you can in the time available.
  - Try pooling information with others (or, if you're up for it, the whole class!)
  - **Attempt to reconstruct the tree** of all messages. This is a big challenge. What sort of strategy might you use? (Hint: write out all the pairwise distances, e.g. in a table. Are any pairs of messages more closely related?)
  - **Do whatever you want.** You can continue to work in large teams, or break up into smaller teams again, or swap people between teams, or try to collect a central database (public or private), or work individually, or any other organizational strategy you wish to try. Leave early if you like. Or stay. Please be respectful and inclusive.

**After the class, submit your best-guess reconstruction of the original (root) sentence to bCourses.**